ORACLE CRM On Demand

Oracle Web Services On Demand Guide

Version 4.0 (Oracle CRM On Demand Release 16)
Rev. A

June 2009

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Oracle CRM On Demand

Chapter 3: Getting Started with Web Services

Chapter 4: Best Practices for Designing Client Applications

Chapter 5: Oracle CRM On Demand Objects Exposed Through Web Services

Chapter 6: Oracle CRM On Demand API Calls

Appendix A: Oracle CRM On Demand API Code Samples

Appendix B: Using Attachments With Web Services On Demand

Index

1 What’s New in This Release

What’s New in Oracle Web Services On Demand Guide, Version 4.0 (Oracle CRM On Demand Release 16) Rev. A

Table 1 lists changes described in this version of the documentation to support Version 4.0 (Oracle CRM On Demand Release 16) Rev. A of the software.

Table 1. What’s New in Oracle Web Services On Demand Guide, Version 4.0 (Oracle CRM On Demand Release 16) Rev. A

Topic	Description
“Logging In to the Web Services Session” on page 28	Information about the URL parameter isEncoded has been added. This is necessary when the login credentials in HTTP headers in login requests contain multi-byte characters.
“Request Rate” on page 33	More information has been added about the situations in which the following error message is displayed: The `maximum rate of requests was exceeded`.
“Using the QueryPage Method” on page 38	Information about best practices for using the QueryPage method has been added, in particular about the use of indexed custom fields.
“The Location Field and Account Records” on page 38	Information about the best practice of specifying isNull for the Location field in Account records has been added.
“Moving Customers Between Pods” on page 39	Information about best practices for reducing impact when moving customers between pods (Oracle CRM On Demand instances) has been added.
“GetMapping” on page 303	Information has been added about the naming conventions for custom objects for the GetMapping method.

What’s New in Oracle Web Services On Demand Guide,Version 4.0 (Oracle CRM On Demand Release 16)

Table 2 lists changes described in this version of the documentation to support Version 4.0 (Oracle CRM On Demand Release 16) of the software.

Table 2. What’s New in Oracle Web Services On Demand Guide, Version 4.0 (Oracle CRM On Demand Release 16)

Topic	Description
“Web Service APIs” on page 16 “Delete (Web Services v2.0)” on page 283 “Execute” on page 284 “Insert (Web Services v2.0)” on page 285 “QueryPage (Web Services v2.0)” on page 286 “Update (Web Services v2.0)” on page 294	Objects can now be accessed through the Web Services v2.0, as well as the Web Services v1.0 API. Chapter 5, “Oracle CRM On Demand Objects Exposed Through Web Services,” has been re-organized into separate sections for Web Services v2.0 and Web Services v1.0. Likewise, Chapter 6, “Oracle CRM On Demand API Calls,” has been re-organized so that there are separate sections for methods relevant to Web Services v2.0 and Web Services v1.0. The Web Services v2.0 API supports the following methods: ■ Delete ■ Execute ■ Insert ■ QueryPage ■ Update
See, for example: “Account” on page 45	For Web Services v1.0 only, the Attachment child object has been exposed for the Account, Campaign, Contact, Custom Object 01-03, Dealer, Lead, Opportunity, Service Request, and Solution parent objects.
“Book” on page 74	For Web Services v1.0 only, you can now manage books through Web Services. Records can be queried, added, deleted or updated for the objects that support books, therefore child objects are exposed for Account, Activity (Task and Appointment), Contact, Custom Object 01-03, Dealer, Household, Lead, Opportunity, Portfolio, and Service Request. The Book parent object has also been exposed.
“Partner” on page 241	For Web Services v2.0 only, the Partner object has been exposed.

Table 2. What's New in Oracle Web Services On Demand Guide, Version 4.0 (Oracle CRM On Demand Release 16)

Topic	Description
"Financial Account" on page 212 "Financial Account Holder" on page 214 "Financial Account Holding" on page 216 "Financial Plan" on page 218 "Financial Product" on page 219 "Financial Transaction" on page 221	For Web Services v2.0 only, the following banking-related objects have been exposed: Financial Account, Financial Account Holder, Financial Account Holding, Financial Plan, Financial Product, and Financial Transaction.
"Claim" on page 196 "Coverage" on page 202 "Damage" on page 208 "Insurance Property" on page 228 "Involved Party" on page 230 "Policy" on page 244 "Policy Holder" on page 246	For Web Services v2.0 only, the following insurance-related objects have been exposed: Claim, Coverage, Damage, Insurance Property, Involved Party, Policy, and Policy Holder.
"QueryPage (Web Services v1.0)" on page 273 "QueryPage (Web Services v2.0)" on page 286	Information has been added about the BookId, BookName, and IncludeSubBooks arguments of the QueryPage method. This provides the capability to filter records by specifying a book.
"GetEvents" on page 301	Information has been added about the new field-level attributes returned by the GetEvents method.
"LoginHistoryQueryPage" on page 307	Administrators can now use the LoginHistoryQueryPage method to query Oracle CRM On Demand for login history information for their company. In previous releases, this information was only available in the Company Administration, Sign In Audit page in the application.
"MergeRecords" on page 308	The MergeRecords method now supports the Household and Portfolio record types.
"UpdatePicklist" on page 311	You can now update, insert, or delete values from picklists that already exist in the Oracle CRM On Demand application. Information has been added about the UpdatePicklist method.

Table 2. What's New in Oracle Web Services On Demand Guide, Version 4.0 (Oracle CRM On Demand Release 16)

Topic	Description
"UpdateCascadingPicklists" on page 312	You can now update existing cascading picklist relationships, or create new cascading picklist relationships for picklists that already exist in the Oracle CRM On Demand application. Information has been added about the UpdateCascadingPicklists method.
"UserUsageQueryPage" on page 314	Administrators can now use the UserUsageQueryPage method to query Oracle CRM On Demand for Web service utilization information. In previous releases, this information was only available in the Web Services Utilization page in the application.
Appendix B, "Using Attachments With Web Services On Demand,"	For Web Services v1.0 only, you can now manage attachments through Web services. URL or file attachments can be added, deleted, or updated for any parent object that supports Attachment as a child object. A new appendix has been added to describe the use of attachments with Web Services On Demand.

Additional Changes

Changes were made to reflect the new branding of Oracle CRM On Demand.

2 Overview of Oracle CRM 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”
- “Web Services and the Oracle CRM On Demand Objects” on page 16
- “Field Types Supported by Oracle CRM On Demand” on page 19
- “Web Services Utilization” on page 24

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 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 fields) 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.

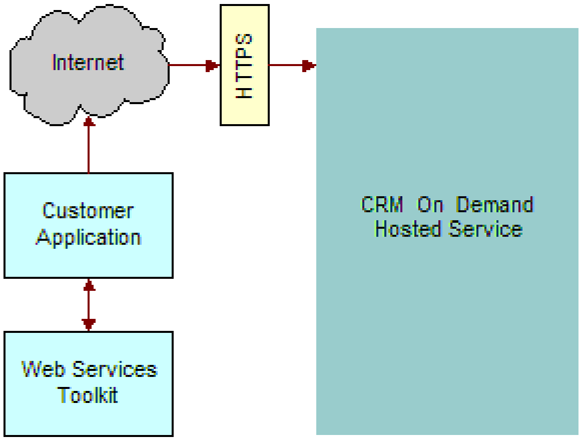

Figure 1. How Web Services Communicate with Oracle CRM On Demand

Oracle CRM On Demand is designed to be backward-compatible with previous releases. Therefore 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:

- 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 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 must be included in all subsequent requests during that session. For more information, see "Establishing and Managing the Web Services Session" on page 27.

Support is provided for part of the UserNameToken profile of the WS-I Basic Security Profile 1.0. The complete profile is not supported, rather only the UserName and Password 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 "Logging In and Making an Integration Request" on page 31.

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

The 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 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 exposed in the Web services, with the names of the Web services matching the default names of the business objects. Chapter 5, "Oracle CRM On Demand Objects Exposed Through Web Services" details the Oracle CRM On Demand parent and child objects that are exposed through Oracle CRM On Demand Web Services.

Web Service APIs

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

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

Before Web Services On Demand Version 4.0, only the Web Services v1.0 was available.

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 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, sortorder, and sortsequence arguments) for issuing queries compared to the Web Services v1.0 QueryPage method.

Table 3 shows the methods available through the Web Services v1.0 and Web Services v2.0 APIs for access to objects:

Table 3. Web Services v1.0 and Web Services v2.0 Methods

Web Services v1.0	Web Services v2.0	Comments
Delete (Web Services v1.0)	Delete (Web Services v2.0)	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	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 (Web Services v1.0)	Insert (Web Services v2.0)	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 (Web Services v1.0)	QueryPage (Web Services v2.0)	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 (Web Services v1.0)	Update (Web Services v2.0)	Updates records with a new value.
UpdateChild	Not applicable	Updates child records with a new value.

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 19.
- 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/`*

Attachment child objects are not exposed for the Web Services v2.0.

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.

To view or modify integration tag information for a record type

1. Navigate to the Field Setup Administration screen 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 screen.
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, refer to 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 should 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 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>&gt;'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 type for fields, so fields are represented by appropriate XSD data types. Table 5 shows the list of supported XSD data types.

Table 5. 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

Table 5. Data Type Mapping in the Web Services v2.0 API

Data Type	Mapped XSD Data Type
ID	xsd:string
NOTE	xsd:string
PHONE	xsd:string
TEXT	xsd:string
INTEGER	xsd:int
TIME	xsd:time
Others	xsd:string

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 exposed 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 exposed. 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>.

The following is a sample request to insert a multi-select picklist field.

```
<?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>

<OpportunityWS_OpportunityInsertOrUpdate_Input xmlns="urn:crmondemand/ws/
opportunity/10/2004">

<ListOfOpportunity>

  <Opportunity>

     <ExternalSystemId>EXT-OPPORTUNITY-000000000MSP1</ExternalSystemId>

     <OpportunityName>OpportunityInsertOrUpdate Insert msp</OpportunityName>

     <SalesStage>Building Vision</SalesStage>

     <Revenue>8000</Revenue>

     <CloseDate>4/15/2005</CloseDate>

     <Forecast>Y</Forecast>

     <Status>Pending</Status>

     <Priority>High</Priority>

     <Probability>40</Probability>

     <msplMS_field>CPL Value 12;CPL Value 13;CPL Value 14</msplMS_field>

  </Opportunity>

</ListOfOpportunity>

</OpportunityWS_OpportunityInsertOrUpdate_Input>

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

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 in Oracle CRM On Demand are in the following format:

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

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 (different locales use different symbols as the decimal point).

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 ().

Web Services Utilization

In the Oracle CRM On Demand application, the Web Services Utilization page provides useful information on Web services usage, both current and historical, for a company.

In the Web Services utilization page, the source of Web services calls is logged to determine whether the calls originated, for example, from a PDA application, or from a generic custom application.

You can also use the UserUsageQueryPage method to get information about Web services utilization. For more information about this method, see "UserUsageQueryPage" on page 314.

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

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:

- "Setting Up Oracle CRM On Demand" on page 25
- "Establishing and Managing the Web Services Session" on page 27

Setting Up 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.**
4. **Establish and Manage the Web Services Session**.

These tasks are described in the following topics.

Requesting Web Services Integration

For security reasons, Web services integration is not automatically enabled for Oracle CRM On Demand customers. On request, an Oracle CRM On Demand Customer Care representative enables the Oracle CRM On Demand Integration capability for your company.

For companies that have Web Services integration enabled, the Web Services Enabled check box is selected in the Company Profile page in the Oracle CRM On Demand application.

Downloading WSDL files and Schema Files

Your company's designated Oracle CRM On Demand administrator accesses the Web Services Administration page in Oracle CRM On Demand to download Web Service Description Language (WSDL) files that have been published for the desired Oracle CRM On Demand objects (record types). The administrator can also download WSDL files for the Service API; the API for managing administrative tasks through Web services.

You can download WSDL files for both the Web Services v1.0 and Web Services v2.0 API, and for both of these services, you can download Custom or Generic WSDL files.

For Web Services v1.0 only, you can also download 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, or Service 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 one of the following buttons, to display a page containing the WSDL. Depending on whether you select Download Custom WSDL, or Download Generic WSDL, custom fields are displayed differently in the WSDL:
   - **Download Custom WSDL.** For Custom WSDL, the XML tags for the custom fields are based on the Integration tags from Field Setup. Custom WSDL allows you to generate WSDL specific to your company that uses your company's field naming conventions.
   - **Download Generic WSDL.** For Generic WSDL, the custom fields are based on generic XML tags: CustomNumber0, CustomCurrency0, and so on. Using these placeholders, together with the Mapping Service allows applications to map to the field names that your company uses.
5. Save the WSDL file to your computer.

Downloading Schema Files

For the Web Services v1.0 API, you may need to download XML Schema files in addition to WSDL files.

The Integration Events Web service (part of the Service API) uses Schema (XSD) files in its WSDL, which you download in an integrationevent.zip file. The .zip file contains the WSDL file and the XSD schema files of all supported record types. However, if you create custom fields or rename fields for a record type, you cannot use the XSD files contained in the integrationevent.zip file for tracking these fields. Instead, you must download an XSD file using the Download Custom Schema button in the Web Services Administration page. For more information about the Integration Events Web service, see "Downloading the Integration Event WSDL File" on page 303.

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.

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.

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.

Establishing and Managing the Web Services Session

A Web services-enabled client, including a client application written in any language that interacts with the Web services framework, must establish a secure session with Oracle CRM On Demand. Throughout this session, 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.

The Web Service API for Oracle CRM On Demand integration is session-based. It has the following features:

- 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. Session management in Oracle CRM On Demand is based on a session ID, JSESSIONID, contained in HTTP Session Cookies. The session ID is critical to successful session generation and maintenance of a SOAP transaction. 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.

  NOTE: If the session ID is used, it must be URL encoded and should not be used directly.

- 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.

SOAP operations do not work if an Oracle Session ID is not referenced in one of these ways.

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.

The logoff operation is considered as one of the operations for a session, so the session ID must be present in the logoff request. However, only the cookie version is accepted in the logoff request.

The Oracle CRM On Demand Web Service interface supports the following three types of function:

- "Logging In to the Web Services Session"
- "Integration Requests for the Web Services Session" on page 31
- "Logging Out of the Web Services Session" on page 32

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

Logging In to the Web Services Session

The login request is an HTTPS request to instantiate a session and obtain a session ID. A client invokes login by sending an HTTP GET request to a URL like the following:

```
https://secure.crmondemand.com/Services/Integration?command=login
```

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

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

The following is an example of an HTTP GET request and headers in clear text without encoding:

```
GET /Services/Integration?command=login HTTP/1.1

UserName: MyUserName

Password: MyPassword

User-Agent: Axis

Host: MyOnDemandServer
```

The following is an example with encoding:

```
GET /Services/Integration?command=login&isEncoded=Y HTTP/1.1

UserName: WS%2f%e6%9c%80%e7%9f%ad%e9%95%bf%e5%ba%a6%e4%b8%ba

Password: MyPassword

User-Agent: Axis

Host: MyOnDemandServer
```

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 logging in, see “Code Samples for Logging In and Logging Out” on page 317.

It is also possible to log in at the same time as making an integration request; for more information, see “Logging In and Making an Integration Request” on page 31.

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://`*`server`*`/Services/Integration?command=ssoitsurl&ssoid=`*`company-sso-id`*

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.

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

For code samples for single sign-on see "Code Samples for Logging In Using Single Sign-On" on page 326.

Outbound SSO

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.

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:

      ```
      https://server/Services/SSOTokenValidate?odSsoToken = "ssotoken value"
      ```

   b. 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:

      ```
      https://server/Services/Integration?command=ssologin&odSsoToken="ssotoken
      value"
      ```

b 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, refer to the Customer Care Portal - Web services resource library.

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.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.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. The following table shows the top-level properties that specify key properties of the HTTP response.

Table 6. 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.

Logging In and Making an Integration Request

It is possible to log into a Web services session without making an explicit log in request. When the first integration request is being sent, you can provide login credentials in the security header along with the request. This is due to support for the UserNameToken profile of the WS-I Basic Security Profile 1.0. In this case, the SOAP header contains a <wsse:UsernameToken> element, which has child elements containing a username and password. The response includes the cookie with a JSESSIONID.

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

A sample SOAP request is as follows:

```
<?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://
schemas.xmlsoap.org/ws/2002/xx/secext" xmlns:xsd="http://www.w3.org/2001/
XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken >
        <wsse:Username>user@emod.com</wsse:Username>
        <wsse:Password>user1285</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
<soap:Body>
    <AccountWS_AccountQueryPage_Input xmlns="urn:crmondemand/ws/account/">
      <PageSize>10</PageSize>
      <ListOfAccount xmlns="urn:/crmondemand/xml/account">
        <Account>
          <AccountId></AccountId>
          <Location></Location>
          <AccountName>LIKE '*'</AccountName>
        </Account>
      </ListOfAccount>
      <StartRowNum>0</StartRowNum>
    </AccountWS_AccountQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

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.crmondemand.com/Services/Integration?command=logoff;
```

NOTE: The parameter value logoff is case sensitive.

Logoff input. 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. The jsessionid returned to the client during login must be included with the request. The request must contain the JSESSIONID as a cookie.

Logoff output. There is no output. If the session does not encounter any errors, the status code of the response is 200. This response indicates that the request succeeded.

For code samples for logging out, see "Code Samples for Logging In and Logging Out" on page 317.

Additional 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.crmondemand.com/Services/Integration;jsessionid=XXXX?command=logoff
```

Limits for the Web Services Session

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.

The limiters are summarized in the following topics; for more information, contact Customer Care.

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:

1. Where the client sends more than 20 requests per second
2. Where the client sends requests at a rate faster than one request per 50 ms.

For the second situation, a wait statement is recommended.

Request Size Limit

The upper limit on the size of any incoming HTTP request is 14 MB. If the request size is exceeded, the following error message is provided to subsequent SOAP requests:

```
Request exceeded the size limit of 14 MB.
```

Session Time-outs

Web services sessions time out after 10 minutes if there is no activity during the session.

Maximum Records Returned

For return messages the maximum number of records returned for each query is limited to 100. If the requested records exceed this limit, an error of the following type is returned:

```
PageSize method argument cannot be greater than 100, specified by the server
parameter ‘MaximumPageSize’.
```

A response never returns more than the specified number of records for a parent object in a request.

Maximum Objects in a Web Services Request

The maximum number of objects that can be sent in a single SOAP request is 20.

4 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:

- "Batch Processing" on page 35
- "Session Management and Pooling" on page 36
- "API Calls" on page 37
- "Moving Customers Between Pods" on page 39
- "Error Handling and Logging" on page 39
- "Handling Outages" on page 40

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 should only be used in instances where longer response time would not impact the user experience. However, for such interactive applications, if the application 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.

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

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

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

The client time-out on a single Web service call should 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 "Establishing and Managing the Web Services Session" on page 27.

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 should send requests in a synchronous manner. If the client application needs to send messages asynchronously, multiple sessions should be used.

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 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 the application 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 should 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.

Using Attachments

For a number of record types, when using the Web Services v1.0, 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 10MB.
- 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 358. 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 B, "Using Attachments With Web Services On Demand,".

Querying Login History and User Usage

When you use the LoginHistoryQueryPage and UserUsageQueryPage methods, the queries should be as specific as possible, that is, you should 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 307 and "UserUsageQueryPage" on page 314.

Using the QueryPage Method

The following are best practices for using the QueryPage method:

- Specify the best operators to make queries faster. For example, use the = operator instead of the * wildcard.
- Specify only fields that are required. Adding all fields or specifying fields that are not required impacts the response time or throughput.
- Use indexed fields for searches. For more information, see the following section.

For more information about the QueryPage method, see "QueryPage (Web Services v1.0)" on page 273, or "QueryPage (Web Services v2.0)" on page 286.

Using Indexed Custom Fields to Improve QueryPage Performance

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 application 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.

The Location Field and Account Records

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. A best practice is therefore to specify isNull for Location instead of blank.

Working with Opportunity Product Revenue Records and Web Services

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.

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 should 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 should 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.

Error Handling and Logging

Error handling and logging are essential when developing a client application. The client application should 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

This topic describes best practices for handling outages due to:

- **Scheduled Maintenance Downtime**. From time to time, Oracle CRM On Demand will have scheduled downtime when the application 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 "Error Handling and Logging" on page 39.

Handling Outages and Failures

The client application should 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.

5 Oracle CRM On Demand Objects Exposed Through Web Services

This chapter contains reference information about the objects exposed 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" on page 41
- "Parent Objects (Web Services v1.0)" on page 44
- "Child Objects (Web Services v1.0)" on page 175
- "Parent Objects (Web Services v2.0)" on page 181

Reference Information About the Parent Objects

The reference information about the parent objects (starting with "Account" on page 45) 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, "Oracle CRM On Demand API Calls."
- The fields that are exposed for the objects:
  - The required and read-only fields
  - The user keys, see "Oracle CRM On Demand User Keys" on page 42
  - The audit fields, see "Audit Fields" on page 42
  - The status key, see "Oracle CRM On Demand Status Keys" on page 43
  - The pick map fields, see "Oracle CRM On Demand Pick Maps" on page 43
  - For Web Services v1.0 only, the filterable fields, see "Filterable Fields" on page 44
  - 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 "Parent Objects (Web Services v1.0)" on page 44, and "Parent Objects (Web Services v2.0)" on page 181.

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, exposed by the Web services API contain the read-only audit fields contained in Table 7.

Table 7. 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.
CreatedDate	The DateTime stamp of when the record was created.

Table 7. Audit Fields for the Oracle CRM On Demand Objects

Field Name	Description
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.

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.

For Web Services v2.0, the Id, CreatedBy, CreatedById, CreatedDate, ModifiedBy, ModifiedById, ModifiedDate and ModId fields are status keys. 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.

The status keys for the Oracle CRM On Demand objects are outlined in "Parent Objects (Web Services v1.0)" on page 44 and "Parent Objects (Web Services v2.0)" on page 181.

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 "Parent Objects (Web Services v1.0)" on page 44 and "Parent Objects (Web Services v2.0)" on page 181.

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 "Parent Objects (Web Services v1.0)" on page 44. For the Web Services V2.0 API, all fields in parent objects are filterable.

Parent Objects (Web Services v1.0)

The following Oracle CRM On Demand objects are detailed in this topic:

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.

Child Components

Activity, Address, Asset, Attachment, Book, Competitor, Contact, CustomObject1 - CustomObject3, Lead, Opportunity, Multiple Contact Roles, Note, PortfolioTeam, Related Account, Revenue, Service Request, and Team

For information about using attachments with this object, see Appendix B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 8 details the methods called by the Account service.

Table 8. Methods Called by Account Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	AccountDelete
“DeleteChild” on page 269	AccountDeleteChild
“Insert (Web Services v1.0)” on page 270	AccountInsert
“InsertChild” on page 271	AccountInsertChild
“InsertOrUpdate” on page 272	AccountInsertOrUpdate
“QueryPage (Web Services v1.0)” on page 273	AccountQueryPage
“Update (Web Services v1.0)” on page 279	AccountUpdate
“UpdateChild” on page 280	AccountUpdateChild

Fields

Table 9 details the required and read-only fields for the account object.

Table 9. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Account	AccountName	Required
	AccountConcatField	Read-only
	Audit Fields	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
	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

Table 9. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only
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

Table 9. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Multiple Contact Roles	ContactRole	Required
	ContactId	Required
	Audit Fields	Read-only
Opportunity	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
RelatedAccount	AccountRelationshipId	Read-only
	RelatedAccountId	Read-only
Revenue	RevenueId	Required
	PartNumber	Required
	Revenue	Required
	ContactFullName	Required
	Audit Fields	Read-only

Table 9. Required and Read-Only Fields for the Account 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 10 details the status key for the account object, and the child component on which this key resides.

Table 10. Status Key for the Account Object

Child Component	Field Name
Account	Audit Fields
	ExternalSystemId
	AccountId
	IntegrationId
	LastUpdated
AccountNote	Audit Fields
	ExternalSystemId
	AccountNoteId
	IntegrationId
AccountTeam	Audit Fields
	AccountTeamId

Table 10. Status Key for the Account Object

Child Component	Field Name
Activity	Audit Fields
	ExternalSystemId
	ActivityId
	IntegrationId
Asset	Audit Fields
	ExternalSystemId
	AssetId
	IntegrationId
Attachment	Audit Fields
	Id
	AccountId
	ModId
Book	Audit Fields
	BookId
	ModId
Competitor	Audit Fields
	AccountCompetitorId
	CompetitorExternalId
	CompetitorId
	CompetitorIntegrationId
Contact	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
CustomObject3	AccountCustomObject3CreatedById
	AccountCustomObject3CreatedDate
	AccountCustomObject3ModifiedById
	AccountCustomObject3ModifiedDate

Table 10. Status Key for the Account Object

Child Component	Field Name
Lead	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
Opportunity	Audit Fields
	AccountId
	ExternalSystemId
	IntegrationId
	OpportunityId
RelatedAccount	Audit Fields
	AccountRelationshipId
Revenue	Audit Fields
	ExternalId
	IntegrationId
	RevenueId
ServiceRequest	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	ServiceRequestId
Partner	Audit Fields
	AccountPartnerId
	PartnerExternalId
	PartnerId
	PartnerIntegrationId
	Updated

Table 11 details the pick map fields for the account object and the child objects on which they reside.

Table 11. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Account	Owner	OwnerId
	ParentAccount, ParentAccountLocation	ParentAccountId
	ParentAccountIntegrationId	ParentAccountId
	ParentAccountExternalSystemId	ParentAccountId
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

Table 11. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Competitor	RelatedAccountExternalId	RelatedAccountId
	RelatedAccountSystemId	RelatedAccountId
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 11. 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 11. 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 11. 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 12 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 12. Filterable Fields and User Key Fields on the Account Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Account	All	AccountId
		IntegrationId
		ExternalSystemId
		AccountName and Location
Account Note	Subject	Subject and Description
Account Team	FirstName	FirstName and Last Name
	LastName	UserID
	UserID	
	UserRole	
	AccountAccess	
	OpportunityAccess	
	ContactAccess	
	ModifiedDate	
	ModifiedByID	

Table 12. 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	ExternalId
	IntegrationId	IntegrationId
	City	
	Country	
	ModifiedDate	
	Province	
	StateProvince	
	ZipCode	
Asset	AssetId	AssetId
	Contract	
	Date	
	ModifiedDate	
	PartNumber	
	ProductCategory	
	Product	
	ProjectManager	
	PurchaseDate	
	Price	
	Quantity	
	SalesRep	
	SerialNumber	
	ShipDate	
	Status	
	Type	
	Warranty	

Table 12. 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	ContactId
	AccountContactModifiedDate	Id
	ContactType	
	ContactFirstName	
	ContactId	
	ContactLastName	
	Id	
	JobTitle	
	Owner	
	RelationshipType	
	RelationshipModifiedDate	
	RelationshipModifiedById	
Competitor	ModifiedDate	CompetitorId
CustomObject3	AccountCustomObject3ModifiedById	CustomObject3Id
	AccountCustomObject3ModifiedDate	ExternalSystemId
	CustomObject3Id	SystemId
	CustomObject3Name	
	ExternalSystemId	
	IntegrationId	
	Type	

Table 12. 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	
	Source	
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
	Id	
Multiple Contact Roles	ContactId	ContactId
	ContactExternalId	ContactIntegrationId
	ContactIntegrationId	ContactExternalId
	ContactRole	
	ModifiedDate	
Opportunity	Owner	OpportunityId
	Revenue	Id
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	Status	
	OpportunityId	
	Id	

Table 12. Filterable Fields and User Key Fields on the Account Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Related Account	AccountRelationshipId	AccountRelationshipId
	Comments	RelatedAccountId
	EndDate	RelatedAccountExternalId
	ModifiedDate	RelatedAccountIntegrationId
	RelatedAccountExternalId	
	RelatedAccountId	
	RelatedAccountIntegrationId	
	RelationshipRole	
	RelationshipStatus	
	RelationshipType	
	ReverseRelationshipRole	
	StartDate	

Table 12. 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	
	Frequency	
	ModifiedDate	
	NumberOfPeriods	
	Product	
	ProductCategoryId	
	ProductCategoryExternalId	
	ProductCategoryIntegrationId	
	ProductExternalId	
	ProductId	
	ProductIntegrationId	
	PurchasePrice	
	Quantity	
	Revenue	
	RevenueId	
	StartCloseDate	
	Status	
	Type	

Table 12. 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	
	Priority	
	Type	
	Cause	
	Source	
	Status	
	Id	
	ServiceRequestId	

Table 13 details the picklists available for the account object.

Table 13. Picklists Available for the Account Object

Child Component	Field Name
Account	AccountType
	Priority
	Region
	CallFrequency
	InfluenceType
	Route
	Status
	MarketPotential
	MarketingSegment
Account Team	TeamRole
Competitor	Role
Multiple Contact Roles	ContactRole
Partner	Role
RelatedAccount	Relationship
	Status

Table 13. Picklists Available for the Account Object

Child Component	Field Name
Revenue	Type
	Status
	Frequency

For more information on the fields exposed through the Account Web service, go to the Web Services Administration screen 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 B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 14 details the methods called by the Activity service.

Table 14. Methods Called by Activity Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	ActivityDelete
“DeleteChild” on page 269	ActivityDeleteChild
“Insert (Web Services v1.0)” on page 270	ActivityInsert
“InsertChild” on page 271	ActivityInsertChild
“InsertOrUpdate” on page 272	ActivityInsertOrUpdate
“QueryPage (Web Services v1.0)” on page 273	ActivityQueryPage
“Update (Web Services v1.0)” on page 279	ActivityUpdate
“UpdateChild” on page 280	ActivityUpdateChild

Fields

Table 15 details the required and read-only fields for the activity object.

Table 15. Required and Read-Only Fields for the Activity Object

Child Component	Field Name	Type
Activity	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 15. 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 16 details the status key for the activity object.

Table 16. Status Key for the Activity Object

Child Component	Field Name
Activity	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
	Id
	UserExternalId
	UserIntegrationId
ProductDetailed	Audit Fields
	ProductDetailedId
	ExternalId
SampleDropped	Audit Fields
	SampleDroppedId
	ExternalId
Solution	Audit Fields
	SolutionId
	ExternalId

Table 17 details the pick map fields for the activity object and the child objects on which they reside.

Table 17. Pick Map Fields for the Activity Object

Child Component	Pick Map Field	Maps To
Activity	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	UserExternalId	UserId
	UserIntegrationId	UserId
SampleDropped	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId

Table 17. 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 18 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 18. Filterable Fields and User Key Fields on the Activity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity	All	ActivityId
		IntegrationId
		ExternalSystemId
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Contact	ContactId	ContactId
	ContactExternalId	ContactExternalId
	ContactIntegrationId	ContactIntegrationId
	ContactFirstName	
	ContactLastName	
	ContactAccountId	
	ContactAccountName	
	ContactAccountLocation	
	ContactAccountIntegrationId	
	ContactAccountExternalSystemId	
	ModifiedDate	

Table 18. Filterable Fields and User Key Fields on the Activity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
User	ModifiedDate	UserId
	UserId	UserExternalId
	UserExternalId	UserIntegrationId
	UserIntegrationId	
	UserEmail	
	UserFirstName	
	UserLastName	
	UserRole	
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	

Table 19 details the picklists available for the activity object.

Table 19. 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 exposed through the Activity Web service, go to the Web Services Administration screen 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 20 details the methods called by the Asset service.

Table 20. Methods Called by Asset Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	AssetDelete
"Insert (Web Services v1.0)" on page 270	AssetInsert
"InsertOrUpdate" on page 272	AssetInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	AssetQueryPage
"Update (Web Services v1.0)" on page 279	AssetUpdate

Fields

Table 21 details the required and read-only fields for the asset object.

Table 21. Required and Read-Only Fields for the Asset Object

Child Component	Field Name	Type
Asset	ProductId	Required
	ProductCategory	Read-only
	PartNumber	Read-only
	Type	Read-only
	Status	Read-only
	Audit Fields	Read-only

Table 22 details the status key for the asset object.

Table 22. Status Key for the Asset Object

Child Component	Field Name
Asset	Audit Fields
	AssetId
	IntegrationId
	ExternalSystemId

Table 23 details the pick map fields for the asset object.

Table 23. Pick Map Fields for the Asset Object

Child Component	Pick Map Field	Maps To
Asset	AccountIntegrationId	AccountId
	AccountExternalSystemId	AccountId
	Account, AccountLocation	AccountId
	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId
	Product	ProductId

Table 24 provides a list of the filterable fields and a list of user key combinations for the asset object.

Table 24. Filterable Fields and User Key Fields on the Asset Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Asset	All	AssetId
		IntegrationId
		ExternalSystemId

Table 25 details the picklists available for the asset object.

Table 25. Picklists Available for the Asset Object

Field Name
Warranty
Contract

For more information on the fields exposed through the Asset Web service, go to the Web Services Administration screen 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 screens. For assistance in gaining access to the Book object, contact your Oracle CRM On Demand service provider.

Parent Objects

Account, Activity, Contact, Dealer, Household, Lead, Opportunity, Portfolio, Service Request.

Child Components

BookUser, SubBook.

Methods Called

Table 26 details the methods called by the Book service.

Table 26. Methods Called by Book Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	BookDelete
“DeleteChild” on page 269	BookDeleteChild
“Insert (Web Services v1.0)” on page 270	BookInsert

Table 26. Methods Called by Book Service

Method	Name as Defined in Service
"InsertChild" on page 271	BookInsertChild
"InsertOrUpdate" on page 272	BookInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	BookQueryPage
"Update (Web Services v1.0)" on page 279	BookUpdate
"UpdateChild" on page 280	BookUpdateChild

Fields

Table 27 details the required and read-only fields for the book object.

Table 27. Required and Read-Only Fields for the Book Object

Child Component	Field Name	Type
Book	BookName	Required
	BookId	Read-only
	Audit Fields	Read-only
BookUser	UserId	Required
	Audit Fields	Read-only
SubBook	UserAlias	Required
	AccessProfileName	Required

Table 28 details the status key for the book object.

Table 28. Status Key for the Book Object

Child Component	Field Name
Book	Audit Fields
	BookId
	BookName
BookUser	Audit Fields
	BookId
SubBook	BookId
	UserId

Table 29 details the pick map fields for the book object.

Table 29. Pick Map Fields for the Book Object

Child Component	Pick Map Field	Maps To
Book	BookName	BookId
BookUser	BookName	BookId
SubBook	BookName	BookId

Table 30 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 30. Filterable Fields and User Key Fields on the Book Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Book	All	BookId
		BookName
BookUser	None	BookId
		UserId
SubBook	None	BookId
		BookName

Table 31 details the picklists available for the book object.

Table 31. Picklists Available for the Book Object

Child Component	Field Name
Book	BookType
	ParentBookName
BookUser	UserAlias
	BookRole
	AccessProfileName

For more information on the fields exposed through the Book Web service, go to the Web Services Administration screen 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 Object

Lead

Child Components

Activity, Attachment, Contact/Recipient, Lead, Opportunity, and Note.

For information about using attachments with this object, see Appendix B, "Using Attachments With Web Services On Demand".

Methods Called

Table 32 details the methods called by the Campaign service.

Table 32. Methods Called by Campaign Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	CampaignDelete
"DeleteChild" on page 269	CampaignDeleteChild
"Insert (Web Services v1.0)" on page 270	CampaignInsert
"InsertChild" on page 271	CampaignInsertChild
"InsertOrUpdate" on page 272	CampaignInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	CampaignQueryPage
"Update (Web Services v1.0)" on page 279	CampaignUpdate
"UpdateChild" on page 280	CampaignUpdateChild

Fields

Table 33 details the required and read-only fields for the campaign object.

Table 33. 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 33. 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 34 details the status key for the campaign object.

Table 34. Status Key for the Campaign Object

Child Component	Field Name
Campaign	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 34. Status Key for the Campaign Object

Child Component	Field Name
Recipient	Audit Fields
	CampaignContactId
	ContactExtrernalId
	ContactIntegrationId
	ContactId

Table 35 details the pick map field for the campaign object.

Table 35. Pick Map Field for the Campaign Object

Child Component	Pick Map Field	Maps To
Campaign	Owner	OwnerId

Table 35. 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 35. 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 36 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 36. Filterable Fields and User Key Fields on the Campaign Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Campaign	All	CampaignId IntegrationId ExternalSystemID
Activity	Type Owner Subject DueDate Priority Status	Type and Description
Attachment	None	Id ExternalSystemId FileNameOrURL and FileExtension
CampaignNote	Subject	Subject and Description
Lead	Campaign EstimatedCloseDate Rating Source Status LeadOwner PotentialRevenue ProductInterest SalesPerson	FirstName Description

Table 36. 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	
	Revenue	
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
Recipient	ContactId	None
	ModifiedDate	

Table 37 details the picklists available for the campaign object.

Table 37. Picklists Available for the Campaign Object

Child Component	Field Name
Campaign	CampaignType
	Status
Contact	DeliveryStatus
	ResponseStatus

For more information on the fields exposed through the Campaign Web service, go to the Web Services Administration screen 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, Household, Opportunity, Portfolio, and Vehicle

Child Components

Account, Activity, Asset, Attachment, Book, Campaign, CustomObject1 - CustomObject3, Interests, Lead, Opportunity, Service Request, Note, Related Contact, Revenue, and Team.

For information about using attachments with this object, see Appendix B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 38 details the methods called by the Contact service.

Table 38. Methods Called by Contact Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	ContactDelete
“DeleteChild” on page 269	ContactDeleteChild
“Insert (Web Services v1.0)” on page 270	ContactInsert
“InsertChild” on page 271	ContactInsertChild
“InsertOrUpdate” on page 272	ContactInsertOrUpdate
“QueryPage (Web Services v1.0)” on page 273	ContactQueryPage
“Update (Web Services v1.0)” on page 279	ContactUpdate
“UpdateChild” on page 280	ContactUpdateChild

Fields

Table 39 details the required and read-only fields for the contact object.

Table 39. Required and Read-Only Fields for the Contact Object

Child Component	Field Name	Type
Contact	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 39. 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 39. 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 39. 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 39. 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 40 details the status key for the contact object.

Table 40. Status Key for the Contact Object

Child Component	Field Name
Contact	Audit Fields
	AccountId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Account	Audit Fields
	Name and Location
	ExternalSystemId
Activity	Audit Fields
	ActivityExternalId
	ActivityId
	ActivityIntegrationId

Table 40. 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 40. 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 40. Status Key for the Contact Object

Child Component	Field Name
Team	Audit Fields
	ContactTeamId
	UserExternalSystemId
	UserId
	UserIntegrationId

Table 41 details the pick map fields for the contact object.

Table 41. Pick Map Fields for the Contact Object

Child Component	Pick Map Field	Maps To
Contact	AccountName	AccountId
	Owner	AssignedToAlias
	SourceCampaignName	SourceCampaignId
	ManagerExternalSystemId	ManagerId
Account	ExternalSystemId	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 41. 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 41. 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 41. 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 42 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 42. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact	All	ContactId
		IntegrationId
		ExternalSystemId

Table 42. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Account	AccountId	AccountId
	ExternalSystemId	ExternalSystemId
	Location	Name and Location
	Name	
Activity	CallType	ActivityIntegrationId
		ActivityID
		ActivityExternalId
Address	AddressID	AddressID
	ExternalId	ExternalId
	IntegrationId	IntegrationId
	City	
	Country	
	ZipCode	
	StateProvince	
	Province	
Asset	AssetId	AssetId
	ExternalSystemId	ExternalSystemId
	IntegrationId	IntegrationId
	ProductId	
	SerialNumber	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 42. Filterable Fields and User Key Fields on the Contact Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Campaign	CampaignContactId	CampaignContactId
	CampaignId	CampaignId
	CampaignExternalSystemId	CampaignExternalSystemId
	CampaignName	CampaignName
	DeliveryStatus	
	ResponseStatus	
	ModifiedDate	
Contact Note	Subject	Subject and Description
Contact Team	ContactTeamId	None
	UserId	
	UserExternalSystemId	
	UserIntegrationId	
	UserFirstName	
	UserLastName	
	ContactAccess	
	UserRole	
CustomObject3	ContactCustomObject3ModifiedById	CustomObject3Id
	ContactCustomObject3ModifiedDate	ExternalSystemId
	CustomObject3Id	SystemId
	CustomObject3Name	
	ExternalSystemId	
	IntegrationId	
	Type	
Interests	InterestId	InterestId
	Category	ExternalSystemId
	Subject	

Table 42. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	Campaign	None
	EstimatedCloseDate	
	Rating	
	Source	
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
	OpportunityId	
Opportunity	Opportunity	None
	OpportunityId	
	Owner	
	Revenue	
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
	Account	

Table 42. Filterable Fields and User Key Fields on the Contact Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Personal Address	PersonalAddressIntegrationId	None
	PersonalAddressName	
	AlternateCity	
	AlternateCountry	
	AlternateZipCode	
	AlternateStateProvince	
	AlternateAddress	
	AlternateAddress2	
	AlternateAddress3	
	Id	
	IntegrationId	
	AddressName	
	City	
	Country	
	ZipCode	
	StateProvince	
	Address	
	ShippingAddress2	
Related Contact	ContactRelationshipId	None
	RelatedContactId	
	RelatedContactExternalId	
	RelatedContactIntegrationId	
	RelationshipStatus	
	RelationshipType	
	StartDate	
	EndDate	
	Description	
	RelationshipRole	
	ReverseRelationshipRole	

Table 42. 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	
	Frequency	
	IntegrationId	
	ModifiedDate	
	NumberOfPeriods	
	Product	
	ProductId	
	ProductCategoryId	
	ProductCategory	
	ProductCategoryExternalId	
	ProductCategoryIntegrationId	
	ProductExternalId	
	ProductIntegrationId	
	PurchasePrice	
	Quantity	
	Revenue	
	RevenueId	
	Status	
	StartCloseDate	
	Type	

Table 42. 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	
	Owner	
	Priority	
	Type	
	Cause	
	Source	
	Status	
	ServiceRequestId	

Table 43 details the picklists available for the contact object.

Table 43. Picklists Available for the Contact Object

Child Component	Field Name
Contact	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 43. Picklists Available for the Contact Object

Child Component	Field Name
Related Contact	Relationship
	Status
Revenue	Type
	Status
	Frequency

Table 44 details a number of contact object fields that you must not use for customer integrations.

Table 44. Contact Object Fields That You Must Not Use

Field Name
PartyTypeCode
PartyUId
PersonUId

For more information on the fields exposed through the Contact Web service, go to the Web Services Administration screen 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 45 details the method called by the Current user service, and its name as defined on the service.

Table 45. Methods Called by Current User Service

Method	Name as Defined in Service
"QueryPage (Web Services v1.0)" on page 273	CurrentUserQueryPage

Fields

Table 46 details the required and read-only fields for the current user object.

Table 46. Required and Read-Only Fields for the Current User Object

Child Component	Field Name	Type
Current User	FirstName	Required
	LastName	Required
	CreatedBy	Read-only
	ModifiedBy	Read-only
Login History	LastLoggedIn	Read-only

Table 47 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 47. Filterable Fields on the Current User Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Current User	All	CurrentUserId
		IntegrationId
		ExternalSystemId
		FirstName and LastName and Middlename
Login History	Alias	None
	SourceIPAddress	
	SignInStatus	
	SignInTime	

For more information on the fields exposed through the Current user Web service, go to the Web Services Administration screen 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 CustomerObject1 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 screen 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.

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 B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 48 details the methods called by the CustomObject1 service. The methods for CustomObject2 and CustomObject3 follow the same pattern.

Table 48. Methods Called by CustomObJect1 Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	CustomObject1Delete
“DeleteChild” on page 269	CustomObject1DeleteChild
“Insert (Web Services v1.0)” on page 270	CustomObject1Insert
“InsertChild” on page 271	CustomObject1InsertChild
“InsertOrUpdate” on page 272	CustomObject1OrUpdate
“QueryPage (Web Services v1.0)” on page 273	CustomObject1Page
“Update (Web Services v1.0)” on page 279	CustomObject1Update
“UpdateChild” on page 280	CustomObject1UpdateChild

Fields

Table 49 details the required and read-only fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 49. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
CustomObject1	CustomObject1Id	Required
	CustomObject1ExternalSystemID	Required
	CustomObject1IntegrationId	Required
	CustomObject1Id	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
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 49. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
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
Portfolio	AccountNumber	Read-only
	CObj1PortfolioCreatedById	Read-only
	CObj1PortfolioCreatedDate	Read-only
	CObj1PortfolioModifiedById	Read-only
	CObj1PortfolioModifiedDate	Read-only
	Revenue	Read-only

Table 49. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
Team	CustomObject1TeamId	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only

Table 50 details the status key for the CustomObject1 object. The status keys for CustomObject2 and CustomObject3 follow a similar pattern.

Table 50. Status Key for the CustomObject1 Object

Child Component	Field Name
CustomObject1	Audit Fields
	CustomObject1Id
	ExternalSystemId
	IntegrationId
Account	CustomObject1AccountId
	CObj1AccountCreatedById
	CObj1AccountCreatedDate
	CObj1AccountModifiedById
	CObj1AccountModifiedDate
Attachment	Audit Fields
	Id
	CustomObjectId
Book	Audit Fields
	BookId
	ModId
Contact	CustomObject1ContactId
	CObj1ContactCreatedById
	CObj1ContactCreatedDate
	CObj1ContactModifiedById
	CObj1ContactModifiedDate

Table 50. Status Key for the CustomObject1 Object

Child Component	Field Name
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 51 details the pick map fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 51. Pick Map Fields for the CustomObject1 Object

Child Component	Pick Map Field	Maps To
CustomObject1	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 51. 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 51. Pick Map Fields for the CustomObject1 Object

Child Component	Pick Map Field	Maps To
Team	UserExternalSystemId	UserId
	UserIntegrationId	UserId
	UserEmail	UserId

Table 52 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 52. Filterable Fields and User Key Fields on the CustomObject1 Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
CustomObject1	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 52. 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 52. 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 53 details the picklists available for the CustomObject1 object.

Table 53. Picklists Available for the CustomObject1 Object

Child Component	Field Name
Dealer	Type
Household	Type
Portfolio	Type
Vehicle	Type

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.

NOTE: To download the Dealer WSDL file, you must be given access to the Dealer object. If you do not have access to the Dealer object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Dealer object, contact your Oracle CRM On Demand service provider.

Child Objects

Attachment and Book.

For information about using attachments with this object, see Appendix B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 54 details the methods called by the Dealer service.

Table 54. Methods Called by Dealer Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	DealerDelete
"DeleteChild" on page 269	DealerDeleteChild
"Insert (Web Services v1.0)" on page 270	DealerInsert
"InsertChild" on page 271	DealerInsertChild
"InsertOrUpdate" on page 272	DealerInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	DealerQueryPage
"Update (Web Services v1.0)" on page 279	DealerUpdate
"UpdateChild" on page 280	DealerUpdateChild

Fields

All fields on the dealer object are filterable.

Table 55 details the required and read-only fields for the dealer object.

Table 55. 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

Table 55. Required and Read-Only Fields for the Dealer Object

Child Component	Field Name	Type
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 56 details the status key for the dealer object.

Table 56. Status Key for the Dealer Object

Child Component	Field Name
Dealer	Audit Fields
	DealerId
	DealerIntegrationID
	DealerExternalSystemId
Attachment	Audit Fields
	Id
	DealerId
Book	Audit Fields
	BookId
	ModId

Table 57 details the pick map fields for the dealer object.

Table 57. Pick Map Fields for the Dealer Object

Child Components	Pick Map Field	Maps To
Dealer	Owner	OwnerId
	ParentDealerExternalSystemId	ParentDealerId
	ParentDealerIntegrationId	ParentDealerId
	ParentDealerName	ParentDealerId
	ParentDealerSite	ParentDealerId
Book	BookName	BookId

Table 58 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 58. Filterable Fields and User Key Fields on the Dealer Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Dealer	All	DealerId
		IntegrationID
		ExternalSystemID
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 59 details the picklists available for the dealer object.

Table 59. Picklists Available for the Dealer Object

Field Name
ParentDealerName
ParentDealerSite

For more information on the fields exposed through the Dealer Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the dealer object.

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.

NOTE: To download the Household WSDL, you must be given access to the Household object. If you do not have access to the Household object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Household object, contact your Oracle CRM On Demand service provider.

Child Component

Book, HouseholdTeam

Methods Called

Table 60 details the methods called by the Household service.

Table 60. Methods Called by Household Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	HouseholdDelete
“DeleteChild” on page 269	HouseholdDeleteChild
“Insert (Web Services v1.0)” on page 270	HouseholdInsert
“InsertChild” on page 271	HouseholdInsertChild
“InsertOrUpdate” on page 272	HouseholdInsertOrUpdate
“QueryPage (Web Services v1.0)” on page 273	HouseholdQueryPage
“Update (Web Services v1.0)” on page 279	HouseholdUpdate
“UpdateChild” on page 280	HouseholdInsertChild

Fields

Table 61 details the required and read-only fields for the household object.

Table 61. Required and Read-Only Fields for the Household Object

Child Component	Field Name	Type
Household	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 61. 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 62 details the status key for the household object.

Table 62. Status Key for the Household Object

Child Component	Field Name
Household	Audit Fields
	ExternalSystemId
	HouseholdId
	IntegrationID
Book	Audit Fields
	BookId
	ModId
Contact	Audit Fields
	ContactId

Table 62. Status Key for the Household Object

Child Component	Field Name
HouseholdTeam	Audit Fields
	UserExternalId
	HouseholdId
	UserAlias
	UserEmail

Table 63 details the pick map fields for the household object.

Table 63. 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 64 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 64. Filterable Fields and User Key Fields on the Household Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Household	All	HouseholdId
		IntegrationID
		ExternalSystemID
Book	None	BookId
		BookName

Table 64. 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	HouseholdId
	UserExternalSystemId	ExternalSystemId
	LastName	UserAlias
	FirstName	UserEmail
	TeamRole	
	HouseholdAccess	
	HouseholdTeamId	

Table 65 details the picklists available for the household object.

Table 65. Picklists Available for the Household Object

Child Component	Field Name
Household	Segment
	Type
Contact	RelationshipRole
HouseholdTeam	TeamRole
	HouseholdAccess

For more information on the fields exposed through the Household Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the household object.

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, and Campaign.

For information about using attachments with this object, see Appendix B, "Using Attachments With Web Services On Demand".

Methods Called

Table 66 details the methods called by the Lead service.

Table 66. Methods Called by Lead Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	LeadDelete
"DeleteChild" on page 269	LeadDeleteChild
"Insert (Web Services v1.0)" on page 270	LeadInsert
"InsertChild" on page 271	LeadInsertChild
"InsertOrUpdate" on page 272	LeadInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	LeadQueryPage
"Update (Web Services v1.0)" on page 279	LeadUpdate
"UpdateChild" on page 280	LeadUpdateChild

Fields

Table 67 details the required and read-only fields for the lead object.

Table 67. Required and Read-Only Fields for the Lead Object

Child Component	Field Name	Type
Lead	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
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

Table 67. Required and Read-Only Fields for the Lead Object

Child Component	Field Name	Type
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 68 details the status key for the lead object.

Table 68. Status Key for the Lead Object

Child Component	Field Name
Lead	Audit Fields
	AccountId
	CampaignId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
	OpportunityId

Table 68. Status Key for the Lead Object

Child Component	Field Name
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LeadId
Attachment	Audit Fields
	Id
	LeadId
Book	Audit Fields
	BookId
	ModId

Table 69 details the pick map fields for the lead object.

Table 69. Pick Map Fields for the Lead Object

Child Component	Pick Map Field	Maps To
Lead	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 69. 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 70 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 70. Filterable Fields and User Key Fields on the Lead Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	All	LeadId
		IntegrationId
		ExternalSystemId
		LeadFirstName and LeadLastName
		Description

Table 70. 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	
	Subject	
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 71 details the picklists available for the lead object.

Table 71. Picklists Available for the Lead Object

Field Name
Country
MrMrs
Rating
Source
StateProvince
Status

For more information on the fields exposed through the Lead Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application and generate the WSDL file for the lead object.

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.

NOTE: To download the MedEd WSDL file, you must be given access to the MedEd object. If you do not have access to the MedEd object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the MedEd object, contact your Oracle CRM On Demand service provider.

Child Component

Invitee

Methods Called

Table 72 details the methods called by the MedEd service.

Table 72. Methods Called by MedEd Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	MedEdDelete
"DeleteChild" on page 269	MedEdDeleteChild
"Insert (Web Services v1.0)" on page 270	MedEdInsert
"InsertChild" on page 271	MedEdInsertChild
"InsertOrUpdate" on page 272	MedEdInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	MedEdQueryPage
"Update (Web Services v1.0)" on page 279	MedEdUpdate
"UpdateChild" on page 280	MedEdUpdateChild

Fields

Table 73 details the read-only fields for the MedEd object and its child component.

Table 73. Read-Only Fields on the MedEd Object

Child Component	Field Name	Type
MedEd	EndDate	Required
	Name	Required
	Objective	Required
	StartDate	Required
	Audit Fields	Read-only
Invitee	InviteeStatus	Required
	InviteeId	Read-only
	Audit Fields	Read-only

Table 74 details the status key for the MedEd object.

Table 74. Status Key for the MedEd Object

Child Component	Field Name
MedEd	Audit Fields
	ExternalId
	MedEdId
Invitee	Audit Fields
	ContactIdExternalId
	MedEdInviteeId

Table 75 details the pickmap fields for the MedEd object and its child objects.

Table 75. Pick Map Fields for the MedEd Object

Child Component	Pick Map Field	Maps To
MedEd	ProductExternalId	ProductId
	ProductIntegrationId	ProductId

Table 76 provides a list of the filterable fields and user key combinations for the child components of the MedEd object.

Table 76. Filterable Fields and User Key Fields on the MedEd Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
MedEd	ExternalSystemId	MedEdId
	ProductIntegrationId	ExternalSystemId
	ProductId	
	ProductId	
	ProductExternalId	
	PrimaryOwnerId	
Invitee	ContactId	MedEdInviteeId
	ContactExternalId	ContactExternalId
	InviteeStatus	
	Type	
	ModifiedDate	

Table 77 details the picklists available for the MedEd object.

Table 77. Picklists Available for the MedEd Object

Child Component	Field Name
MedEd	EventStatusCode
	EventTypeCode
Invitee	InviteeStatus

For more information on the fields exposed through the MedEd Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the MedEd object.

Related Topic

Invitee

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 78 details the methods called by the Note service.

Table 78. Methods Called by Note Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	NoteDelete
"Insert (Web Services v1.0)" on page 270	NoteInsert
"InsertOrUpdate" on page 272	NoteInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	NoteQueryPage
"Update (Web Services v1.0)" on page 279	NoteUpdate

Fields

Table 79 details the required and read-only fields for the note object.

Table 79. Required and Read-Only Fields for the Note Object

Child Component	Field Name	Type
Note	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 80 details the status key for the note object.

Table 80. Status Key for the Note Object

Child Component	Field Name
Note	Audit Fields
	NoteId

Table 81 provides a list of the filterable fields and a list of user key combinations for the note object.

Table 81. Filterable Fields and User Key Fields on the Note Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Note	All	NoteId

For more information on the fields exposed through the Note Web service, go to the Web Services Administration screen 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

Child Components

Activity, Attachment, Book, Contact, Competitor, Lead, OpportunityTeam, and Note.

NOTE: The Revenue child object for Opportunity is actually called Product.

For information about using attachments with this object, see Appendix B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 82 details the methods called by the Opportunity service.

Table 82. Methods Called by Opportunity Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	OpportunityDelete
“DeleteChild” on page 269	OpportunityDeleteChild
“Insert (Web Services v1.0)” on page 270	OpportunityInsert
“InsertChild” on page 271	OpportunityInsertChild
“InsertOrUpdate” on page 272	OpportunityInsertOrUpdate
“QueryPage (Web Services v1.0)” on page 273	OpportunityQueryPage
“Update (Web Services v1.0)” on page 279	OpportunityUpdate
“UpdateChild” on page 280	OpportunityUpdateChild

Fields

Table 83 details the required and read-only fields for the opportunity object.

Table 83. Required and Read-Only Fields for the Opportunity Object

Child Component	Field Name	Type
Opportunity	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 83. 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 83. 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 84 details the status key for the opportunity object.

Table 84. Status Key for the Opportunity Object

Child Component	Field Name
Opportunity	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 84. 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 85 details the pick map fields for the opportunity object.

Table 85. Pick Map Fields for the Opportunity Object

Child Component	Pick Map Field	Maps To
Opportunity	Owner	OwnerId
	AccountExternalSystemId	AccountId
	Territory	TerritoryId
	KeyContactIntegrationId	KeyContactId
	KeyContactExternalSystemId	KeyContactId

Table 85. 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 85. 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

Table 86 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 86. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Opportunity	All	OpportunityId
		IntegrationId
		ExternalSystemId
Activity	Type	Type and Description
	Owner	
	Subject	
	DueDate	
	Priority	
	Status	

Table 86. Filterable Fields and User Key Fields on the Opportunity Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Competitor	OpportunityCompetitorId	OpportunityCompetitorId
	PrimaryContactId	ExternalSystemId
	EndDate	CompetitorId
	CompetitorId	
	CompetitorExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	
Contact	ContactType	AccountName and Private
	ContactFirstName	ContactFirstName and ContactLastName and Private
	JobTitle	
	ContactLastName	
	Owner	
	Id	
Lead	Campaign	None
	EstimatedCloseDate	
	Rating	
	Source	
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	

Table 86. Filterable Fields and User Key Fields on the Opportunity Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Note	Subject	Subject and Description
Partner	OpportunityPartnerId	OpportunityPartnerId
	PrimaryContactId	ExternalSystemId
	EndDate	PartnerId
	PartnerId	
	PartnerExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	
Product	OpportunityIntegrationId	OpportunityExternalSystemId
	OpportunityExternalSystemId	OpportunityIntegrationId
	OpportunitySalesStage	ProductRevenueId
	OpportunityAccountId	ExternalId
	OpportunityAccountName	IntegrationId
	OpportunityAccountLocation	
	OpportunityAccountExternalSystemId	
	OpportunityAccountIntegrationId	
	ModifiedDate	
	ContactId	
	ContactExternalSystemId	
	ContactIntegrationId	
	Contract	
	OwnerId	
	Owner	
	SerialNumber	
	Revenue	
	ExpectedRevenue	
	Quantity	
	PurchasePrice	
	PurchaseDate	

Table 86. 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	
	ExternalId	
	IntegrationId	
	ProductId	
	ProductName	
	ProductExternalSystemID	
	ProductionIntegrationId	
	ProductCategoryId	
	ProductCategory	
	ProductPartNumber	
	ProductStatus	
	ProductType	
	OpportunityId	
	OpportunityName	

Table 87 details the picklists available for the opportunity object.

Table 87. Picklists Available for the Opportunity Object

Child Component	Field Name
Opportunity	LeadSource
	Priority
	Probability
	ReasonWonLost
	Status
	Type
	Year
	Make
	Model
OpportunityTeam	TeamRole
Product	Frequency
	Probability
	Status
	Type
	Warranty
	Contract

For more information on the fields exposed through the Opportunity Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the opportunity 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.

NOTE: To download the Portfolio WSDL file, you must be given access to the Portfolio object. If you do not have access to the Portfolio object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Portfolio object, contact your Oracle CRM On Demand service provider.

Child Component

Book, Contact, PortfolioTeam

Methods Called

Table 88 details the methods called by the Portfolio service.

Table 88. Methods Called by Portfolio Service

Method	Name as Defined in Service
“Delete (Web Services v1.0)” on page 266	PortfolioDelete
“DeleteChild” on page 269	PortfolioDeleteChild
“Insert (Web Services v1.0)” on page 270	PortfolioInsert
“InsertChild” on page 271	PortfolioInsertChild
“InsertOrUpdate” on page 272	PortfolioInsertOrUpdate
“QueryPage (Web Services v1.0)” on page 273	PortfolioQueryPage
“Update (Web Services v1.0)” on page 279	PortfolioUpdate
“UpdateChild” on page 280	PortfolioUpdateChild

Fields

Table 89 details the read-only fields for the portfolio object and its child component.

Table 89. Read-Only Fields on the Portfolio Object

Child Component	Field Name	Type
Portfolio	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 89. 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 90 details the status key for the portfolio object.

Table 90. Status Key for the Portfolio Object

Child Component	Field Name
Portfolio	Audit Fields
	ExternalSystemId
	PortfolioId
	IntegrationId
Book	Audit Fields
	BookId
	ModId
Contact	Audit Fields
	ContactId
	Id
PortfolioTeam	Audit Fields
	UserId
	UserAlias
	UserEmail
	UserExternalSystemId

Table 91 details the pickmap fields for the portfolio object and its child objects.

Table 91. Pick Map Fields for the Portfolio Object

Child Component	Pick Map Field	Maps To
Portfolio	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 92 provides a list of the filterable fields and user key combinations for the child components of the portfolio object.

Table 92. Filterable Fields and User Key Fields on the Portfolio Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Portfolio	All	PortfolioId
		IntegrationId
		ExternalSystemId
Book	None	BookId
		BookName

Table 92. 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 93 details the picklists available for the portfolio object.

Table 93. Picklists Available for the Portfolio Object

Child Component	Field Name
Portfolio	AccountType
	Status
	TermUnit
Contact	Relationship
PortfolioTeam	TeamRole
	PortfolioAccess

For more information on the fields exposed through the Portfolio Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the portfolio object.

Related Topic

Fields

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 94 details the methods called by the Product service.

Table 94. Methods Called by Product Service

Method	Name as Defined in Service
"Insert (Web Services v1.0)" on page 270	ProductInsert
"InsertOrUpdate" on page 272	ProductInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	ProductQueryPage
"Update (Web Services v1.0)" on page 279	ProductUpdate

Fields

All fields on the product object are filterable.

Table 95 details the required and read-only fields for the product object.

Table 95. Required and Read-Only Fields for the Product Object

Child Component	Field Name	Type
Product	ProductName	Required
	Audit Fields	Read-only

Table 96 details the status key for the product object.

Table 96. Status Key for the Product Object

Child Component	Field Name
Product	Audit Fields
	Id
	IntegrationId

Table 97 details the pick map field for the product object.

Table 97. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 98 details the user keys for the product object.

Table 98. User Keys for the Product Object

Child Component	Field Name
Product	ProductId
	IntegrationId
	ExternalSystemId

Table 99 details the picklists available for the product object.

Table 99. Picklists Available for the Product Object

Field Name
BodyStyle
Category
Class
CurrencyCode
DoorStyle
Engine

Table 99. Picklists Available for the Product Object

Field Name
Make
Model
PriceType
ProductType
Revision
Status
SubType
TherapeuticClass
Transmission
Trim

For more information on the fields exposed through the Product Web service, go to the Web Services Administration screen 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.

Methods Called

Table 100 details the methods called by the Product category service.

Table 100. Methods Called by Product Category Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	ProductCategoryDelete
"Insert (Web Services v1.0)" on page 270	ProductCategoryInsert
"InsertOrUpdate" on page 272	ProductCategoryInsertOrUpdate

Table 100. Methods Called by Product Category Service

Method	Name as Defined in Service
"QueryPage (Web Services v1.0)" on page 273	ProductCategoryQueryPage
"Update (Web Services v1.0)" on page 279	ProductCategoryUpdate

Fields

All fields on the product category object are filterable.

Table 101 details the required and read-only fields for the product category object.

Table 101. 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 102 details the status key for the product category object.

Table 102. Status Key for the Product Category Object

Child Component	Field Name
ProductCategory	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	Name

Table 103 details the pick map field for the product category object.

Table 103. Pick Map Field for the Product Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 104 details the user keys for the product category object.

Table 104. User Keys for the Product Category Object

Child Component	Field Name
ProductCategory	ProductCategoryId
	IntegrationId
	ExternaSystemId
	Name

For more information on the fields exposed through the Product category Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the product category object.

Related Topic

Product

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, Book, Solution, Audit Trail and Note.

For information about using attachments with this object, see Appendix B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 105 details the methods called by the Service request service.

Table 105. Methods Called by Service Request Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	ServiceRequestDelete
"DeleteChild" on page 269	ServiceRequestDeleteChild
"Insert (Web Services v1.0)" on page 270	ServiceRequestInsert
"InsertChild" on page 271	ServiceRequestInsertChild
"InsertOrUpdate" on page 272	ServiceRequestInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	ServiceRequestQueryPage
"Update (Web Services v1.0)" on page 279	ServiceRequestUpdate
"UpdateChild" on page 280	ServiceRequestUpdateChild

Fields

Table 106 details the required and read-only fields for the service request object.

Table 106. Required and Read-Only Fields for the Service Request Object

Child Component	Field Name	Type
ServiceRequest	ContactEmail	Read-only
	ContactFirstName	Read-only
	ContactFullName	Read-only
	ContactLastName	Read-only
	Audit Fields	Read-only
	LastUpdated	Read-only
	ServiceRequestConcatId	Read-only

Table 106. Required and Read-Only Fields for the Service Request Object

Child Component	Field Name	Type
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 107 details the status key for the service request object.

Table 107. Status Key for the Service Request Object

Child Component	Field Name
ServiceRequest	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId

Table 107. Status Key for the Service Request Object

Child Component	Field Name
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 108 details the pick map fields for the service request object.

Table 108. Pick Map Fields for the Service Request Object

Child Component	Pick Map Field	Maps To
Service Request	Owner	OwnerId
	AccountExternalSystemId	AccountId
	AssetIntegrationId	AssetId
	AssetExternalSystemId	AssetId
	ProductExternalSystemId	ProductId
Book	BookName	BookId

Table 109 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 109. Filterable Fields and User Key Fields on the Service Request Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Service Request	All	ServiceRequestId IntegrationId ExternalSystemId SRNumber
Activity	Type Owner Subject DueDate Priority Status	Type and Description
Attachment	None	Id ExternalSystemId FileNameOrURL and FileExtension
Audit Trail	Date User FieldModified	None
Book	None	BookId BookName
Service Request Note	Subject	Subject and Description
Solution	Title Published SolutionId Status Id	Title

Table 110 details the picklists available for the service request object.

Table 110. Picklists Available for the Service Request Object

Field Name
Area
Cause
Priority
Source
Status
Type

For more information on the fields exposed through the Service request Web service, go to the Web Services Administration screen 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 B, “Using Attachments With Web Services On Demand”.

Methods Called

Table 111 details the methods called by the Solution service.

Table 111. Methods Called by Solution Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	SolutionDelete
"DeleteChild" on page 269	SolutionDeleteChild
"Insert (Web Services v1.0)" on page 270	SolutionInsert
"InsertChild" on page 271	SolutionInsertChild
"InsertOrUpdate" on page 272	SolutionInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	SolutionQueryPage
"Update (Web Services v1.0)" on page 279	SolutionUpdate
"UpdateChild" on page 280	SolutionUpdateChild

Fields

Table 112 details the required and read-only fields for the solution object.

Table 112. Required and Read-Only Fields for the Solution Object

Child Component	Field Name	Type
Solution	Title	Required
	Audit Fields	Read-only
	CreatorId	Read-only
	LastUpdated	Read-only
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 113 details the status key for the solution object.

Table 113. Status Key for the Solution Object

Child Component	Field Name
Solution	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Attachment	Audit Fields
	Id
	SolutionId
ServiceRequest	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 114 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 114. Filterable Fields and User Key Fields on the Solution Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Solution	All	SolutionId
		IntegrationId
		ExternalSystemId
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension

Table 114. Filterable Fields and User Key Fields on the Solution Object’s Child Components

Child Components	Filterable Fields	User Key Field Combinations
Service Request	Subject	SRNumber
	Area	
	Owner	
	Priority	
	Type	
	Cause	
	Source	
	Status	

Table 115 details the picklists available for the solution object.

Table 115. Picklists Available for the Solution Object

Field Name
Area
Cause
Priority
Source
Status
Type

For more information on the fields exposed through the Solution Web service, go to the Web Services Administration screen 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.

Methods Called

Table 116 details the methods called by the Territory service.

Table 116. Methods Called by Territory Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	TerritoryDelete
"Insert (Web Services v1.0)" on page 270	TerritoryInsert
"InsertOrUpdate" on page 272	TerritoryInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	TerritoryQueryPage
"Update (Web Services v1.0)" on page 279	TerritoryUpdate

Fields

All fields on the territory object are filterable. The TerritoryName field is a user key for the territory object.

Table 117 details the required and read-only fields for the territory object.

Table 117. Required and Read-Only Fields for the Territory Object

Child Component	Field Name	Type
Territory	TerritoryName	Required
	Territory	Read-only
	Audit Fields	Read-only

Table 118 details the status key for the territory object.

Table 118. Status Key for the Territory Object

Child Component	Field Name
Territory	Audit Fields
	ExternalSystemId
	IntegrationId
	TerritoryId

Table 119 details the pick map field for the territory object.

Table 119. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

Table 120 details the picklists available for the territory object.

Table 120. Picklists Available for the Territory Object

Field Name
ParentTerritoryExternalSystemId
ParentTerritoryIntegrationId

For more information on the fields exposed through the Territory Web service, go to the Web Services Administration screen 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 the application, 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

Methods Called

Table 121 details the methods called by the User service.

Table 121. Methods Called by User Service

Method	Name as Defined in Service
"DeleteChild" on page 269	UserDeleteChild
"Insert (Web Services v1.0)" on page 270	UserInsert
"InsertChild" on page 271	UserInsertChild
"InsertOrUpdate" on page 272	UserInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	UserQueryPage
"Update (Web Services v1.0)" on page 279	UserUpdate
"UpdateChild" on page 280	UserUpdateChild

Fields

All fields on the user object are filterable.

Table 122 details the required and read-only fields for the user object.

Table 122. 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 123 details the status key for the user object.

Table 123. Status Key for the User Object

Child Component	Field Name
User	ModifiedById
	ModifiedDate
	EMailAddr
	UserId
	IntegrationId

Table 124 details the pick map field for the user object.

Table 124. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

Table 125 provides a list of user key combinations for the user object.

Table 125. User Key Fields on the User Object

Child Components	User Key Field Combinations
User	UserId
	ExternalSystemId
	IntegrationId
	EmailAddr

For more information on the fields exposed through the User Web service, go to the Web Services Administration screen 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. Userscan only be a member of one group, and groups can contain many users.

Child Component

User

Methods Called

Table 126 details the methods called by the User group service.

Table 126. Methods Called by User Group Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	UserGroupDelete
"DeleteChild" on page 269	UserGroupDeleteChild
"Insert (Web Services v1.0)" on page 270	UserGroupInsert
"InsertChild" on page 271	UserGroupInsertChild
"InsertOrUpdate" on page 272	UserGroupInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	UserGroupQueryPage
"Update (Web Services v1.0)" on page 279	UserGroupUpdate
"UpdateChild" on page 280	UserGroupUpdateChild

Fields

Table 127 details the required and read-only fields for the user group object.

Table 127. Required and Read-Only Fields for the User Group Object

Child Component	Field Name	Type
User Group	Name	Required
	UserGroupId	Read-only
	Audit Fields	Read-only

Table 127. Required and Read-Only Fields for the User Group Object

Child Component	Field Name	Type
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 128 details the status key for the user group object.

Table 128. Status Key for the User Group Object

Child Component	Field Name
UserGroup	Audit Fields
	UserGroupId
	UserGroupIntegrationId
	UserGroupExternalSystemId
User	Audit Fields
	Members_UserId
	UserExternalSystemId
	UserIntegrationId

Table 129 details the pick map field for the user group object.

Table 129. Pick Map Field for the User Group Object

Child Component	Pick Map Field	Maps To
User	UserIntegrationId	UserId
	UserExternalSystemId	UserId

Table 130 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 130. Filterable Fields and User Key Fields on the User Group Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
User Group	All	Name
User	UserGroupId	None
	UserId	
	UserIntegrationId	
	UserExternalSystemId	
	Alias	
	Email	
	Role	
	UserFirstName	
	UserLastName	

For more information on the fields exposed through the User group Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the user group 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.

NOTE: To download the Vehicle WSDL file, you must be given access to the Vehicle object. If you do not have access to the Vehicle object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Vehicle object, contact your Oracle CRM On Demand service provider.

Child Component

Contact

Methods Called

Table 131 details the methods called by the Vehicle service.

Table 131. Methods Called by Vehicle Service

Method	Name as Defined in Service
"Delete (Web Services v1.0)" on page 266	VehicleDelete
"DeleteChild" on page 269	VehicleDeleteChild
"Insert (Web Services v1.0)" on page 270	VehicleInsert
"InsertChild" on page 271	VehicleInsertChild
"InsertOrUpdate" on page 272	VehicleInsertOrUpdate
"QueryPage (Web Services v1.0)" on page 273	VehicleQueryPage
"Update (Web Services v1.0)" on page 279	VehicleUpdate
"UpdateChild" on page 280	VehicleUpdateChild

Fields

Table 132 details the required and read-only fields for the vehicle object.

Table 132. Required and Read-Only Fields for the Vehicle Object

Child Component	Field Name	Type
Vehicle	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 133 details the status key for the vehicle object.

Table 133. Status Key for the Vehicle Object

Child Component	Field Name
Vehicle	Audit Fields
	ExternalSystemId
	IntegrationId
	VehicleId
Contact	Audit Fields
	ContactId

Table 134 details the pick map fields for the vehicle object.

Table 134. Pick Map Fields for the Vehicle Object

Child Component	Pick Map Field	Maps To
Vehicle	AccountName	AccountId
	AccountSite	AccountId
	AccountIntegrationId	AccountId
	AccountExternalID	AccountId
	SellingDealerExternalId	SellingDealerId
	SellingDealerIntegrationId	SellingDealerId
	ServicingDealerExternalId	ServicingDealerId
	ServicingDealerIntegrationId	ServicingDealerId
Contact	ContactExternalSystemId	ContactId
	ContactIntegrationId	ContactId

Table 135 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 135. Filterable Fields and User Key Fields on the Vehicle Object’s Child Components

Child Component	Filterable Fields	User Key Field Combinations
Vehicle	All	VehicleId
		ExternalSystemId
		IntegrationId
Contact	ContactId	ContactId
	ContactExternalSystemId	ContactExternalSystemId
	ContactIntegrationId	ContactIntegrationId
	ContactFirstName	

Table 136 details the picklists available for the vehicle object.

Table 136. 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 exposed through the Vehicle Web service, go to the Web Services Administration screen 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" on page 175
- "Attachment" on page 176
- "Audit Trail" on page 176
- "BookUser" on page 176
- "Competitor" on page 176
- "HouseholdTeam" on page 177
- "Interests" on page 177
- "Invitee" on page 177
- "Login History" on page 178
- "Multiple Contact Roles" on page 178
- "OpportunityTeam" on page 178
- "PortfolioTeam" on page 178
- "ProductsDetailed" on page 179
- "Recipient" on page 179
- "Revenue" on page 180
- "Related Account" on page 179
- "Related Contact" on page 180
- "SampleDropped" on page 180
- "SubBook" on page 180
- "Team" on page 181

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 Object

Account, Contact

Attachment

The attachment object stores information about a file or URL that is attached to a record in the application.

Parent Object

Account, Activity, Campaign, Contact, CustomObject1 - CustomObject3, Dealer, 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 exposed, go to the Web Services Administration screen 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 137 details the picklists available for the competitor object.

Table 137. Picklists Available for the Competitor Object

Field Name
RelationshipRole
ReverseRelationshipRole

For more information on the competitor fields exposed, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the account object.

Parent Object

Account, Opportunity

HouseholdTeam

The HouseholdTeam object stores the information on a team that shares household records.

For more information on the household team fields exposed, go to the Web Services Administration screen 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 Objects

Account

OpportunityTeam

The OpportunityTeam object stores information about a team that shares opportunity records.

For more information on the opportunity team fields exposed, go to the Web Services Administration screen 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 exposed, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the portfolio object.

Parent Object

Portfolio

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 exposed, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Parent Object

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 Object

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 exposed, go to the Web Services Administration screen 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 exposed, go to the Web Services Administration screen 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 Object

Account and Contact

Parent Objects (Web Services v2.0)

The following Oracle CRM On Demand objects are detailed in this topic:

- "Account" on page 182
- "Activity" on page 184
- "Asset" on page 188
- "Book" on page 191
- "Campaign" on page 193
- "Category" on page 195
- "Claim" on page 196
- "Contact" on page 199
- "Coverage" on page 202
- "CustomObject" on page 205
- "Damage" on page 208
- "Dealer" on page 210
- "Financial Account" on page 212
- "Financial Account Holder" on page 214
- "Financial Account Holding" on page 216
- "Financial Plan" on page 218
- "Financial Product" on page 219
- "Financial Transaction" on page 221
- "Group" on page 223
- "Household" on page 225
- "Insurance Property" on page 228
- "Involved Party" on page 230

- “Lead” on page 232
- “MedEd” on page 235
- “Note” on page 237
- “Opportunity” on page 239
- “Partner” on page 241
- “Policy” on page 244
- “Policy Holder” on page 246
- “Portfolio” on page 248
- “Product” on page 251
- “Service Request” on page 253
- “Solution” on page 255
- “Territory” on page 257
- “User” on page 259
- “Vehicle” on page 261

Objects exposed 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.

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.

Methods Called

Table 138 details the methods called by the Account service.

Table 138. Methods Called by Account Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	AccountDelete
“Execute” on page 284	AccountExecute
“Insert (Web Services v2.0)” on page 285	AccountInsert
“QueryPage (Web Services v2.0)” on page 286	AccountQueryPage
“Update (Web Services v2.0)” on page 294	AccountUpdate

Fields

Table 139 details the required and read-only fields for the account object.

Table 139. Required and Read-Only Fields for the Account Object

Field Name	Type
AccountName	Required
AccountConcatField	Read-only
Audit Fields	Read-only

Table 140 details the status key for the account object.

Table 140. Status Key for the Account Object

Field Name
Audit Fields
ExternalSystemId
AccountId
IntegrationId
LastUpdated

Table 141 details the pick map fields for the account object.

Table 141. Pick Map Fields for the Account Object

Pick Map Field	Maps To
Owner	OwnerId
ParentAccount, ParentAccountLocation	ParentAccountId
ParentAccountIntegrationId	ParentAccountId
ParentAccountExternalSystemId	ParentAccountId

Table 142 provides a list of the a list of the user key combinations for the account object.

Table 142. User Key Fields on the Account Object

User Key Field Combinations
AccountId
IntegrationId
ExternalSystemId
AccountName and Location

Table 143 details the picklists available for the account object.

Table 143. Picklists Available for the Account Object

Field Name
AccountType
Priority
Region
CallFrequency
InfluenceType
Route
Status
MarketPotential
MarketSegment

For more information on the fields exposed through the Account Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the account object.

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

Methods Called

Table 144 details the methods called by the Activity service.

Table 144. Methods Called by Activity Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	ActivityDelete
"Execute" on page 284	ActivityExecute
"Insert (Web Services v2.0)" on page 285	ActivityInsert
"QueryPage (Web Services v2.0)" on page 286	ActivityQueryPage
"Update (Web Services v2.0)" on page 294	ActivityUpdate

Fields

Table 145 details the required and read-only fields for the activity object.

Table 145. 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

Table 145. Required and Read-Only Fields for the Activity Object

Field Name	Type
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

Table 146 details the status key for the activity object.

Table 146. Status Key for the Activity Object

Field Name
Audit Fields
ActivityId
ExternalSystemId
IntegrationId

Table 147 details the pick map fields for the activity object.

Table 147. Pick Map Fields for the Activity Object

Pick Map Field	Maps To
AccountName	AccountId
AccountLocation	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
Owner	OwnerId
CampaignExternalSystemId	CampaignId
CampaignIntegrationId	CampaignId

Table 147. Pick Map Fields for the Activity Object

Pick Map Field	Maps To
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

Table 148 provides a list of the user key combinations for the activity object.

Table 148. User Key Fields on the Activity Object

User Key Field Combinations
ActivityId
IntegrationId
ExternalSystemId

Table 149 details the picklists available for the activity object.

Table 149. Picklists Available for the Activity Object

Field Name
AccountName
AccountLocation
AccountIntegrationId

Table 149. Picklists Available for the Activity Object

Field Name
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 exposed through the Activity Web service, go to the Web Services Administration screen 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.

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 150 details the methods called by the Asset service.

Table 150. Methods Called by Asset Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	AssetDelete
“Execute” on page 284	AssetExecute
“Insert (Web Services v2.0)” on page 285	AssetInsert
“QueryPage (Web Services v2.0)” on page 286	AssetQueryPage
“Update (Web Services v2.0)” on page 294	AssetUpdate

Fields

Table 151 details the required and read-only fields for the asset object.

Table 151. 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 152 details the status key for the asset object.

Table 152. Status Key for the Asset Object

Field Name
Audit Fields
AssetId

Table 152. Status Key for the Asset Object

Field Name
IntegrationId
ExternalSystemId

Table 153 details the pick map fields for the asset object.

Table 153. 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 154 provides a list of the user key combinations for the asset object.

Table 154. User Key Fields on the Asset Object

User Key Field Combinations
AssetId
IntegrationId
ExternalSystemId

Table 155 details the picklists available for the asset object.

Table 155. Picklists Available for the Asset Object

Field Name
Warranty
Contract

For more information on the fields exposed through the Asset Web service, go to the Web Services Administration screen 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 screens. For assistance in gaining access to the Book object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 156 details the methods called by the Book service.

Table 156. Methods Called by Book Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	BookDelete
"Execute" on page 284	BookExecute
"Insert (Web Services v2.0)" on page 285	BookInsert
"QueryPage (Web Services v2.0)" on page 286	BookQueryPage
"Update (Web Services v2.0)" on page 294	BookUpdate

Fields

Table 157 details the required and read-only fields for the book object.

Table 157. Required and Read-Only Fields for the Book Object

Field Name	Type
BookName	Required
BookId	Read-only
Audit Fields	Read-only

Table 158 details the status key for the book object.

Table 158. Status Key for the Book Object

Field Name
Audit Fields
BookId
BookName

Table 159 details the pick map fields for the book object.

Table 159. Pick Map Fields for the Book Object

Pick Map Field	Maps To
BookName	BookId

Table 160 provides a list of the user key combinations for the book object.

Table 160. User Key Fields on the Book Object

User Key Field Combinations
BookId
BookName

Table 161 details the picklists available for the book object.

Table 161. Picklists Available for the Book Object

Field Name
BookType
ParentBookName

For more information on the fields exposed through the Book Web service, go to the Web Services Administration screen 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.

Methods Called

Table 162 details the methods called by the Campaign service.

Table 162. Methods Called by Campaign Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	CampaignDelete
"Execute" on page 284	CampaignExecute
"Insert (Web Services v2.0)" on page 285	CampaignInsert
"QueryPage (Web Services v2.0)" on page 286	CampaignQueryPage
"Update (Web Services v2.0)" on page 294	CampaignUpdate

Fields

Table 163 details the required and read-only fields for the campaign object.

Table 163. 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 164 details the status key for the campaign object.

Table 164. Status Key for the Campaign Object

Field Name
Audit Fields
ExternalSystemId

Table 164. Status Key for the Campaign Object

Field Name
Id
IntegrationId

Table 165 details the pick map field for the campaign object.

Table 165. Pick Map Field for the Campaign Object

Pick Map Field	Maps To
Owner	OwnerId

Table 166 provides a list of the user key combinations for the campaign object.

Table 166. User Key Fields on the Campaign Object

User Key Field Combinations
CampaignId
IntegrationId
ExternalSystemID

Table 167 details the picklists available for the campaign object.

Table 167. Picklists Available for the Campaign Object

Field Name
CampaignType
Status

For more information on the fields exposed through the Campaign Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application and generate the WSDL file for the campaign object.

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 168 details the methods called by the Category service.

Table 168. Methods Called by Product Category Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	CategoryDelete
"Execute" on page 284	CategoryExecute
"Insert (Web Services v2.0)" on page 285	CategoryInsert
"QueryPage (Web Services v2.0)" on page 286	CategoryQueryPage
"Update (Web Services v2.0)" on page 294	CategoryUpdate

Fields

Table 169 details the required and read-only fields for the category object.

Table 169. Required and Read-Only Fields for the Category Object

Field Name	Type
CategoryName	Required
Audit Fields	Read-only
ModifiedByFullName	Read-only

Table 170 details the status key for the category object.

Table 170. Status Key for the Category Object

Field Name
Audit Fields
ExternalSystemId
Id

Table 170. Status Key for the Category Object

Field Name
IntegrationId
Name

Table 171 details the pick map field for the category object.

Table 171. Pick Map Field for the Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 172 details the user keys for the category object.

Table 172. User Keys for the Category Object

Field Name
ProductCategoryId
IntegrationId
ExternaSystemId
Name

For more information on the fields exposed through the Category Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the category object.

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.

NOTE: To download the Claim WSDL file, you must be given access to the claim object. If you do not have access to the claim object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the claim object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 173 details the methods called by the Claim service.

Table 173. Methods Called by Claim Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	ClaimDelete
"Execute" on page 284	ClaimExecute
"Insert (Web Services v2.0)" on page 285	ClaimInsert
"QueryPage (Web Services v2.0)" on page 286	ClaimQueryPage
"Update (Web Services v2.0)" on page 294	ClaimUpdate

Fields

Table 174 details the required and read-only fields for the claim object.

Table 174. 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 175 details the status key for the claim object.

Table 175. Status Key for the Claim Object

Field Name
Audit Fields
ExternalSystemId
ClaimNumber
IntegrationId

Table 176 details the pickmap fields for the claim object.

Table 176. Pick Map Fields for the Claim Object

Pick Map Field	Maps To
ExternalId	ClaimNumber
IntegrationId	ClaimNumber

Table 177 provides a list of the user key combinations for the claim object.

Table 177. User Key Fields on the Claim Object

User Key Field Combinations
ClaimNumber
IntegrationId
ExternalSystemId

Table 178 details the picklists available for the claim object.

Table 178. Picklists Available for the Claim Object

Field Name
AbilitytoWork
AtFault
CategoryofLoss
ClassofEmployee
InjuredDuringWorkingHours
KindofLoss
LiabilitySource
LineofBusiness
LocationofLoss
LossCode
LossType
MedicalInjuryCode
PartofBodyInjured

Table 178. Picklists Available for the Claim Object

Field Name
PlaceofInjury
RelationshiptoInsured
ReportedBy
State
Status
TypeofInjury

For more information on the fields exposed through the Claim Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the claim object.

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.

Methods Called

Table 179 details the methods called by the Contact service.

Table 179. Methods Called by Contact Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	ContactDelete
"Execute" on page 284	ContactExecute
"Insert (Web Services v2.0)" on page 285	ContactInsert
"QueryPage (Web Services v2.0)" on page 286	ContactQueryPage
"Update (Web Services v2.0)" on page 294	ContactUpdate

Fields

Table 180 details the required and read-only fields for the contact object.

Table 180. 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 181 details the status key for the contact object.

Table 181. Status Key for the Contact Object

Field Name
Audit Fields
AccountId
ExternalSystemId
Id
IntegrationId
LastUpdated

Table 182 details the pick map fields for the contact object.

Table 182. Pick Map Fields for the Contact Object

Pick Map Field	Maps To
AccountName	AccountId
Owner	AssignedToAlias

Table 182. Pick Map Fields for the Contact Object

Pick Map Field	Maps To
SourceCampaignName	SourceCampaignId
ManagerExternalSystemId	ManagerId

Table 183 provides a list of the user key combinations for the contact object.

Table 183. User Key Fields on the Contact Object

User Key Field Combinations
ContactId
IntegrationId
ExternalSystemId

Table 184 details the picklists available for the contact object.

Table 184. Picklists Available for the Contact Object

Field Name
ContactType
LeadSource
MrMrs
BestTimeToCall
CallFrequency
CurrentInvestmentMix
Degree
ExperienceLevel
Gender
InvestmentHorizon
LifeEvent
MaritalStatus
MarketPotential
Objective
OwnOrRent

Table 184. Picklists Available for the Contact Object

Field Name
PrimaryGoal
RiskProfile
Route
Segment
Tier

Table 185 details a number of contact object fields that you must not use for customer integrations.

Table 185. Contact Object Fields That You Must Not Use

Field Name
PartyTypeCode
PartyUId
PersonUId

For more information on the fields exposed through the Contact Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the contact object.

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.

NOTE: To download the Coverage WSDL file, you must be given access to the coverage object. If you do not have access to the coverage object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the coverage object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 186 details the methods called by the Coverage service.

Table 186. Methods Called by Coverage Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	CoverageDelete
"Execute" on page 284	CoverageExecute
"Insert (Web Services v2.0)" on page 285	CoverageInsert
"QueryPage (Web Services v2.0)" on page 286	CoverageQueryPage
"Update (Web Services v2.0)" on page 294	CoverageUpdate

Fields

Table 187 details the required and read-only fields for the coverage object.

Table 187. 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 188 details the status key for the coverage object.

Table 188. Status Key for the Coverage Object

Field Name
Audit Fields
ExternalSystemId
CoverageName
IntegrationId

Table 189 details the pickmap fields for the coverage object.

Table 189. Pick Map Fields for the Coverage Object

Pick Map Field	Maps To
ExternalId	CoverageName
IntegrationId	CoverageName

Table 190 provides a list of the user key combinations for the coverage object.

Table 190. User Key Fields on the Coverage Object

User Key Field Combinations
CoverageName
IntegrationId
ExternalSystemId

Table 191 details the picklists available for the coverage object.

Table 191. Picklists Available for the Coverage Object

Field Name
Coverage
Status

For more information on the fields exposed through the Coverage Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the coverage object.

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 CustomerObject1 is given as an example.

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 screen or available to use Web service calls. For assistance in gaining access to the CustomObject objects, contact your Oracle CRM On Demand service provider.

Methods Called

Table 192 details the methods called by the CustomObject1 service. The methods for the other Custom Objects follow the same pattern.

Table 192. Methods Called by CustomObJect1 Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	CustomObject1Delete
“Execute” on page 284	CustomObject1Execute
“Insert (Web Services v2.0)” on page 285	CustomObject1Insert
“QueryPage (Web Services v2.0)” on page 286	CustomObject1QueryPage
“Update (Web Services v2.0)” on page 294	CustomObject1Update

Fields

Table 193 details the required and read-only fields for the CustomObject1 object.

Table 193. Required and Read-Only Fields for the CustomObject1 Object

Field Name	Type
CustomObject1Id	Required
CustomObject1ExternalSystemID	Required
CustomObject1IntegrationId	Required
CustomObject1Id	Read-only

Table 194 details the status key for the CustomObject1 object.

Table 194. Status Key for the CustomObject1 Object

Field Name
Audit Fields
CustomObject1Id
ExternalSystemId
IntegrationId

Table 195 details the pick map fields for the CustomObject1 object.

Table 195. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
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
CustomObject*n*ExternalId	CustomObjectnId
CustomObject*n*IntegrationId	CustomObject*n*Id
CustomObject*n*Name	CustomObject*n*Id
DealerName	DealerId
HouseholdExternalId	HouseholdId

Table 195. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
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
SolutionIntegrationId	SolutionId
SolutionTitle	SolutionId
ServiceRequestExternalId	ServiceRequestId
ServiceRequestIntegrationId	ServiceRequestId
ServiceRequestName	ServiceRequestId
VIN	VehicleId

Table 196 provides a list of the user key combinations for the CustomObject1 object.

Table 196. User Key Fields on the CustomObject1 Object

User Key Field Combinations
CustomObject1Id

Table 196. User Key Fields on the CustomObject1 Object

User Key Field Combinations
ExternalSystemId
IntegrationId

Damage

The damage object allows you to define and record details about the damages to properties covered in an insurance policy.

NOTE: To download the Damage WSDL file, you must be given access to the damage object. If you do not have access to the damage object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the damage object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 197 details the methods called by the Damage service.

Table 197. Methods Called by Damage Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	DamageDelete
"Execute" on page 284	DamageExecute
"Insert (Web Services v2.0)" on page 285	DamageInsert
"QueryPage (Web Services v2.0)" on page 286	DamageQueryPage
"Update (Web Services v2.0)" on page 294	DamageUpdate

Fields

Table 198 details the required and read-only fields for the damage object.

Table 198. Read-Only Fields on the Damage Object

Field Name	Type
ClaimNumber	Required
DamageName	Required
CurrencyCode	Read-only
Owner	Read-only

Table 198. Read-Only Fields on the Damage Object

Field Name	Type
OwnerId	Read-only
PrimaryContact	Read-only
Audit Fields	Read-only

Table 199 details the status key for the damage object.

Table 199. Status Key for the Damage Object

Field Name
Audit Fields
ExternalSystemId
DamageId
IntegrationId

Table 200 details the pickmap fields for the damage object.

Table 200. Pick Map Fields for the Damage Object

Pick Map Field	Maps To
ExternalId	DamageName
IntegrationId	DamageName

Table 201 provides a list of the user key combinations for the damage object.

Table 201. User Key Fields on the Damage Object

User Key Field Combinations
DamageName
IntegrationId
ExternalSystemId

For more information on the fields exposed through the Damage Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the damage object.

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.

NOTE: To download the Dealer WSDL file, you must be given access to the Dealer object. If you do not have access to the Dealer object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Dealer object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 202 details the methods called by the Dealer service.

Table 202. Methods Called by Dealer Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	DealerDelete
"Execute" on page 284	DealerExecute
"Insert (Web Services v2.0)" on page 285	DealerInsert
"QueryPage (Web Services v2.0)" on page 286	DealerQueryPage
"Update (Web Services v2.0)" on page 294	DealerUpdate

Fields

Table 203 details the required and read-only fields for the dealer object.

Table 203. 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 204 details the status key for the dealer object.

Table 204. Status Key for the Dealer Object

Field Name
Audit Fields
DealerId
DealerIntegrationID
DealerExternalSystemId

Table 205 details the pick map fields for the dealer object.

Table 205. Pick Map Fields for the Dealer Object

Pick Map Field	Maps To
Owner	OwnerId
ParentDealerExternalSystemId	ParentDealerId
ParentDealerIntegrationId	ParentDealerId
ParentDealerName	ParentDealerId
ParentDealerSite	ParentDealerId

Table 206 provides a list of the user key combinations for the Dealer object.

Table 206. User Key Fields on the Dealer Object

User Key Field Combinations
DealerId
IntegrationID
ExternalSystemID

Table 207 details the picklists available for the dealer object.

Table 207. Picklists Available for the Dealer Object

Field Name
ParentDealerName
ParentDealerSite

For more information on the fields exposed through the Dealer Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the dealer object.

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.

Methods Called

Table 208 details the methods called by the Financial Account service.

Table 208. Methods Called by Financial Account Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	FinancialAccountDelete
"Execute" on page 284	FinancialAccountExecute
"Insert (Web Services v2.0)" on page 285	FinancialAccountInsert
"QueryPage (Web Services v2.0)" on page 286	FinancialAccountQueryPage
"Update (Web Services v2.0)" on page 294	FinancialAccountUpdate

Fields

Table 209 details the required and read-only fields for the financial account object.

Table 209. Required and Read-Only Fields for the Financial Account Object

Field Name	Type
FinancialAccount	Required
FinancialAccountNumber	Required

Table 209. Required and Read-Only Fields for the Financial Account Object

Field Name	Type
Type	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 210 details the status key for the financial account object.t

Table 210. Status Key for the Financial Account Object

Field Name
Audit Fields
ExternalSystemId
FinancialAccount
IntegrationId

Table 211 details the pick map fields for the financial account object.

Table 211. Pick Map Fields for the Financial Account Object

Pick Map Field	Maps To
IntegrationId	FinancialAccount
ExternalSystemId	FinancialAccount
ParentFinancialAccountIntegrationId	ParentFinancialAccountId
ParentFinancialAccountExternalSystemId	ParentFinancialAccountId

Table 212 provides a list of the a list of the user key combinations for the financial account object.

Table 212. User Key Fields on the Financial Account Object

User Key Field Combinations
FinancialAccount
IntegrationId
ExternalSystemId

Table 213 details the picklists available for the financial account object.

Table 213. Picklists Available for the Financial Account Object

Field Name
Type
HomeBranch

For more information on the fields exposed through the Financial Account Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the financial account object.

Financial Account Holder

The financial account holder object stores information about the 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.

Methods Called

Table 214 details the methods called by the Financial Account Holder service.

Table 214. Methods Called by Financial Account Holder Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	FinancialAccountHolderDelete
"Execute" on page 284	FinancialAccountHolderExecute
"Insert (Web Services v2.0)" on page 285	FinancialAccountHolderInsert
"QueryPage (Web Services v2.0)" on page 286	FinancialAccountHolderQueryPage
"Update (Web Services v2.0)" on page 294	FinancialAccountHolderUpdate

Fields

Table 215 details the required and read-only fields for the financial account holder object.

Table 215. Required and Read-Only Fields for the Financial Account Holder Object

Field Name	Type
FinancialAccountHolderName	Required
FinancialAccount	Required

Table 215. Required and Read-Only Fields for the Financial Account Holder Object

Field Name	Type
Role	Required
CurrrencyCode	Read-only
Audit Fields	Read-only

Table 216 details the status key for the financial account holder object.

Table 216. Status Key for the Financial Account Holder Object

Field Name
Audit Fields
ExternalSystemId
FinancialAccountHolderName
IntegrationId

Table 217 details the pick map fields for the financial account holder object.

Table 217. Pick Map Fields for the Financial Account Object

Pick Map Field	Maps To
IntegrationId	FinancialAccountHolderName
ExternalSystemId	FinancialAccountHolderName

Table 218 provides a list of the a list of the user key combinations for the financial account holder object.

Table 218. User Key Fields on the Financial Account Holder Object

User Key Field Combinations
FinancialAccountHolderName
IntegrationId
ExternalSystemId

Table 219 details the picklists available for the financial account holder object.

Table 219. Picklists Available for the Financial Account Holder Object

Field Name
Role

For more information on the fields exposed through the Financial Account Holder Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the financial account holder object.

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.

Methods Called

Table 220 details the methods called by the Financial Account Holding service.

Table 220. Methods Called by Financial Account Holding Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	FinancialAccountHoldingDelete
"Execute" on page 284	FinancialAccountHoldingExecute
"Insert (Web Services v2.0)" on page 285	FinancialAccountHoldingInsert
"QueryPage (Web Services v2.0)" on page 286	FinancialAccountHoldingQueryPage
"Update (Web Services v2.0)" on page 294	FinancialAccountHoldingUpdate

Fields

Table 221 details the required and read-only fields for the financial account holding object.

Table 221. Required and Read-Only Fields for the Financial Account Holding Object

Field Name	Type
FinancialAccountHoldingsName	Required
FinancialAccount	Required
FinancialProductFinancialProductName	Required

Table 221. Required and Read-Only Fields for the Financial Account Holding Object

Field Name	Type
CurrencyCode	Read-only
Audit Fields	Read-only

Table 222 details the status key for the financial account holding object.t

Table 222. Status Key for the Financial Account Holding Object

Field Name
Audit Fields
ExternalSystemId
FinancialAccountHoldingsName
IntegrationId

Table 223 details the pick map fields for the financial account holding object.

Table 223. Pick Map Fields for the Financial Account Holding Object

Pick Map Field	Maps To
IntegrationId	FinancialAccountHoldingsName
ExternalSystemId	FinancialAccountHoldingsName

Table 224 provides a list of the a list of the user key combinations for the financial account holding object.

Table 224. User Key Fields on the Financial Account Holding Object

User Key Field Combinations
FinancialAccountHoldingsName
IntegrationId
ExternalSystemId

For more information on the fields exposed through the Financial Account Holding Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the financial account holding object.

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.

Methods Called

Table 225 details the methods called by the Financial Plan service.

Table 225. Methods Called by Financial Plan Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	FinancialPlanDelete
"Execute" on page 284	FinancialPlanExecute
"Insert (Web Services v2.0)" on page 285	FinancialPlanInsert
"QueryPage (Web Services v2.0)" on page 286	FinancialPlanQueryPage
"Update (Web Services v2.0)" on page 294	FinancialPlanUpdate

Fields

Table 226 details the required and read-only fields for the financial plan object.

Table 226. 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 227 details the status key for the financial plan object.t

Table 227. Status Key for the Financial Plan Object

Field Name
Audit Fields
ExternalSystemId

Table 227. Status Key for the Financial Plan Object

Field Name
FinancialPlan
IntegrationId

Table 228 details the pick map fields for the financial plan object.

Table 228. Pick Map Fields for the Financial Plan Object

Pick Map Field	Maps To
IntegrationId	FinancialPlan
ExternalSystemId	FinancialPlan

Table 229 provides a list of the a list of the user key combinations for the financial plan object.

Table 229. User Key Fields on the Financial Plan Object

User Key Field Combinations
FinancialPlan
IntegrationId
ExternalSystemId

Table 230 details the picklists available for the financial plan object.

Table 230. Picklists Available for the Financial Plan Object

Field Name
Type
Status

For more information on the fields exposed through the Financial Plan Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the financial plan object.

Financial Product

The financial product object stores information about the products and services that a financial institution offers to its customers.

Methods Called

Table 231 details the methods called by the Financial Product service.

Table 231. Methods Called by Financial Product Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	FinancialProductDelete
"Execute" on page 284	FinancialProductExecute
"Insert (Web Services v2.0)" on page 285	FinancialProductInsert
"QueryPage (Web Services v2.0)" on page 286	FinancialProductQueryPage
"Update (Web Services v2.0)" on page 294	FinancialProductUpdate

Fields

Table 232 details the required and read-only fields for the financial product object.

Table 232. Required and Read-Only Fields for the Financial Product Object

Field Name	Type
FinancialProductName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

Table 233 details the status key for the financial product object.t

Table 233. Status Key for the Financial Product Object

Field Name
Audit Fields
ExternalSystemId
FinancialProductName
IntegrationId

Table 234 details the pick map fields for the financial product object.

Table 234. Pick Map Fields for the Financial Product Object

Pick Map Field	Maps To
IntegrationId	FinancialProductName
ExternalSystemId	FinancialProductName
ParentFinancialProductIntegrationId	ParentFinancialProductId
ParentFinancialProductExternalSystemId	ParentFinancialProductId

Table 235 provides a list of the a list of the user key combinations for the financial product object.

Table 235. User Key Fields on the Financial Product Object

User Key Field Combinations
FinancialProductName
IntegrationId
ExternalSystemId

Table 236 details the picklists available for the financial product object.

Table 236. Picklists Available for the Financial Product Object

Field Name
Category
Type
Class
Sub-Class

For more information on the fields exposed through the Financial Product Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the financial product object.

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.

Methods Called

Table 237 details the methods called by the Financial Transaction service.

Table 237. Methods Called by Financial Transaction Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	FinancialTransactionDelete
“Execute” on page 284	FinancialTransactionExecute
“Insert (Web Services v2.0)” on page 285	FinancialTransactionInsert
“QueryPage (Web Services v2.0)” on page 286	FinancialTransactionQueryPage
“Update (Web Services v2.0)” on page 294	FinancialTransactionUpdate

Fields

Table 238 details the required and read-only fields for the financial transaction object.

Table 238. 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 239 details the status key for the financial transaction object.t

Table 239. Status Key for the Financial Transaction Object

Field Name
Audit Fields
ExternalSystemId
TransactionID
IntegrationId

Table 240 details the pick map fields for the financial transaction object.

Table 240. Pick Map Fields for the Financial Transaction Object

Pick Map Field	Maps To
IntegrationId	TransactionID
ExternalSystemId	TransactionID
FinancialTransactionParentIntegrationId	FinancialTransactionParentId
FinancialTransactionParentExternalSystemId	FinancialTransactionParentId

Table 241 provides a list of the a list of the user key combinations for the financial transaction object.

Table 241. User Key Fields on the Financial Transaction Object

User Key Field Combinations
TransactionID
IntegrationId
ExternalSystemId

Table 242 details the picklists available for the financial transaction object.

Table 242. Picklists Available for the Financial Transaction Object

Field Name
TransactionType

For more information on the fields exposed through the Financial Transaction Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the financial transaction object.

Group

The Group object allows you to create groups to which users can be added. Userscan 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 243 details the methods called by the Group service.

Table 243. Methods Called by Group Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	GroupDelete
"Execute" on page 284	GroupExecute
"Insert (Web Services v2.0)" on page 285	GroupInsert
"QueryPage (Web Services v2.0)" on page 286	GroupQueryPage
"Update (Web Services v2.0)" on page 294	GroupUpdate

Fields

Table 244 details the required and read-only fields for the group object.

Table 244. Required and Read-Only Fields for the Group Object

Field Name	Type
Name	Required
UserGroupId	Read-only
Audit Fields	Read-only

Table 245 details the status key for the group object.

Table 245. Status Key for the Group Object

Field Name
Audit Fields
UserGroupId
UserGroupIntegrationId
UserGroupExternalSystemId

Table 246 details the pick map field for the group object.

Table 246. Pick Map Field for the Group Object

Pick Map Field	Maps To
UserIntegrationId	UserId
UserExternalSystemId	UserId

Table 247 provides a list of the user key combinations for the group object.

Table 247. User Key Fields on the Group Object

User Key Field Combinations
Name

For more information on the fields exposed through the Group Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the group object.

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.

NOTE: To download the Household WSDL, you must be given access to the Household object. If you do not have access to the Household object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Household object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 248 details the methods called by the Household service.

Table 248. Methods Called by Household Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	HouseholdDelete
“Execute” on page 284	HouseholdExecute

Table 248. Methods Called by Household Service

Method	Name as Defined in Service
"Insert (Web Services v2.0)" on page 285	HouseholdInsert
"QueryPage (Web Services v2.0)" on page 286	HouseholdQueryPage
"Update (Web Services v2.0)" on page 294	HouseholdUpdate

Fields

Table 249 details the required and read-only fields for the household object.

Table 249. Required and Read-Only Fields for the Household Object

Field Name	Type
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

Table 249. Required and Read-Only Fields for the Household Object

Field Name	Type
PrimaryGoal	Read-only
Audit Fields	Read-only

Table 250 details the status key for the household object.

Table 250. Status Key for the Household Object

Field Name
Audit Fields
ExternalSystemId
HouseholdId
IntegrationID

Table 251 details the pick map fields for the household object.

Table 251. Pick Map Field for the Household Object

Pick Map Field	Maps To
ContactExternalId	ContactId
ContactIntegrationId	

Table 252 provides a list of the user key combinations for the Household object.

Table 252. User Key Fields on the Household Object

User Key Field Combinations
HouseholdId
IntegrationID
ExternalSystemID

Table 253 details the picklists available for the household object.

Table 253. Picklists Available for the Household Object

Field Name
Segment
Type

For more information on the fields exposed through the Household Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the household object.

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.

NOTE: To download the Insurance Property WSDL file, you must be given access to the insurance property object. If you do not have access to the insurance property object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the insurance property object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 254 details the methods called by the Insurance Property service.

Table 254. Methods Called by Insurance Property Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	InsurancePropertyDelete
“Execute” on page 284	InsurancePropertyExecute
“Insert (Web Services v2.0)” on page 285	InsurancePropertyInsert
“QueryPage (Web Services v2.0)” on page 286	InsurancePropertyQueryPage
“Update (Web Services v2.0)” on page 294	InsurancePropertyUpdate

Fields

Table 255 details the required and read-only fields for the insurance property object.

Table 255. 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 256 details the status key for the insurance property object.

Table 256. Status Key for the Insurance Property Object

Field Name
Audit Fields
ExternalSystemId
PolicyId
IntegrationId

Table 257 details the pickmap fields for the insurance property object.

Table 257. Pick Map Fields for the Insurance Property Object

Pick Map Field	Maps To
ExternalId	PolicyId
IntegrationId	PolicyId

Table 258 provides a list of the user key combinations for the insurance property object.

Table 258. User Key Fields on the Insurance Property Object

User Key Field Combinations
PolicyId
IntegrationId
ExternalSystemId

Table 259 details the picklists available for the insurance property object.

Table 259. Picklists Available for the Insurance Property Object

Field Name
Type

For more information on the fields exposed through the Insurance Property Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the insurance property object.

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.

NOTE: To download the Involved Party WSDL file, you must be given access to the involved party object. If you do not have access to the involved party object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the involved party object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 260 details the methods called by the Involved Party service.

Table 260. Methods Called by Involved Party Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	InvolvedPartyDelete
“Execute” on page 284	InvolvedPartyExecute

Table 260. Methods Called by Involved Party Service

Method	Name as Defined in Service
“Insert (Web Services v2.0)” on page 285	InvolvedPartyInsert
“QueryPage (Web Services v2.0)” on page 286	InvolvedPartyQueryPage
“Update (Web Services v2.0)” on page 294	InvolvedPartyUpdate

Fields

Table 261 details the required and read-only fields for the involved party object.

Table 261. 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 262 details the status key for the involved party object.

Table 262. Status Key for the Involved Party Object

Field Name
Audit Fields
ExternalSystemId
InvolvedPartyName
IntegrationId

Table 263 details the pickmap fields for the involved party object.

Table 263. Pick Map Fields for the Involved Party Object

Pick Map Field	Maps To
ExternalId	InvolvedPartyName
IntegrationId	InvolvedPartyName

Table 264 provides a list of the user key combinations for the involved party object.

Table 264. User Key Fields on the Involved Party Object

User Key Field Combinations
InvolvedPartyName
IntegrationId
ExternalSystemId

Table 265 details the picklists available for the involved party object.

Table 265. Picklists Available for the Involved Party Object

Field Name
Role
RoleinAccident
Location

For more information on the fields exposed through the Involved Party Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the involved party object.

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.

Methods Called

Table 266 details the methods called by the Lead service.

Table 266. Methods Called by Lead Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	LeadDelete
"Execute" on page 284	LeadExecute
"Insert (Web Services v2.0)" on page 285	LeadInsert
"QueryPage (Web Services v2.0)" on page 286	LeadQueryPage
"Update (Web Services v2.0)" on page 294	LeadUpdate

Fields

Table 267 details the required and read-only fields for the lead object.

Table 267. Required and Read-Only Fields for the Lead Object

Field Name	Type
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 268 details the status key for the lead object.

Table 268. Status Key for the Lead Object

Field Name
Audit Fields
AccountId
CampaignId

Table 268. Status Key for the Lead Object

Field Name
ContactId
ExternalSystemId
Id
IntegrationId
LastUpdated
OpportunityId

Table 269 details the pick map fields for the lead object.

Table 269. 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 270 provides a list of the user key combinations of the lead object.

Table 270. User Key Fields on the Lead Object

User Key Field Combinations
LeadId
IntegrationId
ExternalSystemId
LeadFirstName and LeadLastName
Description

Table 271 details the picklists available for the lead object.

Table 271. Picklists Available for the Lead Object

Field Name
Country
MrMrs
Rating
Source
StateProvince
Status

For more information on the fields exposed through the Lead Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application and generate the WSDL file for the lead object.

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.

NOTE: To download the MedEd WSDL file, you must be given access to the MedEd object. If you do not have access to the MedEd object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the MedEd object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 272 details the methods called by the MedEd service.

Table 272. Methods Called by MedEd Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	MedEdDelete
“Execute” on page 284	MedEdExecute
“Insert (Web Services v2.0)” on page 285	MedEdInsert
“QueryPage (Web Services v2.0)” on page 286	MedEdQueryPage
“Update (Web Services v2.0)” on page 294	MedEdUpdate

Fields

Table 273 details the read-only fields for the MedEd object.

Table 273. Read-Only Fields on the MedEd Object

Field Name	Type
EndDate	Required
Name	Required
Objective	Required
StartDate	Required
Audit Fields	Read-only

Table 274 details the status key for the MedEd object.

Table 274. Status Key for the MedEd Object

Field Name
Audit Fields
ExternalId
MedEdId

Table 275 details the pickmap fields for the MedEd object.

Table 275. Pick Map Fields for the MedEd Object

Pick Map Field	Maps To
ProductExternalId	ProductId
ProductIntegrationId	ProductId

Table 276 provides a list of the user key combinations for the MedEd object.

Table 276. User Key Fields on the MedEd Object

Filterable Fields	User Key Field Combinations
ExternalSystemId	MedEdId

Table 276. User Key Fields on the MedEd Object

Filterable Fields	User Key Field Combinations
ProductIntegrationId	ExternalSystemId
ProductId	
ProductId	
ProductExternalId	
PrimaryOwnerId	

Table 277 details the picklists available for the MedEd object.

Table 277. Picklists Available for the MedEd Object

Field Name
EventStatusCode
EventTypeCode

For more information on the fields exposed through the MedEd Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the MedEd 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.

Methods Called

Table 278 details the methods called by the Note service.

Table 278. Methods Called by Note Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	NoteDelete
"Execute" on page 284	NoteExecute
"Insert (Web Services v2.0)" on page 285	NoteInsert

Table 278. Methods Called by Note Service

Method	Name as Defined in Service
"QueryPage (Web Services v2.0)" on page 286	NoteQueryPage
"Update (Web Services v2.0)" on page 294	NoteUpdate

Fields

Table 279 details the required and read-only fields for the note object.

Table 279. 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 280 details the status key for the note object.

Table 280. Status Key for the Note Object

Field Name
Audit Fields
NoteId

Table 281 provides a list of the user key combinations for the note object.

Table 281. User Key Fields on the Note Object

User Key Field Combinations
NoteId

For more information on the fields exposed through the Note Web service, go to the Web Services Administration screen 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.

Methods Called

Table 282 details the methods called by the Opportunity service.

Table 282. Methods Called by Opportunity Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	OpportunityDelete
"Execute" on page 284	OpportunityExecute
"Insert (Web Services v2.0)" on page 285	OpportunityInsert
"QueryPage (Web Services v2.0)" on page 286	OpportunityQueryPage
"Update (Web Services v2.0)" on page 294	OpportunityUpdate

Fields

Table 283 details the required and read-only fields for the opportunity object.

Table 283. 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 284 details the status key for the opportunity object.

Table 284. Status Key for the Opportunity Object

Field Name
Audit Fields
AccountId
ExternalSystemId
Id
IntegrationId

Table 285 details the pick map fields for the opportunity object.

Table 285. Pick Map Fields for the Opportunity Object

Pick Map Field	Maps To
Owner	OwnerId
AccountExternalSystemId	AccountId
Territory	TerritoryId
KeyContactIntegrationId	KeyContactId
KeyContactExternalSystemId	KeyContactId

Table 286 provides a list of the user key combinations for the Opportunity object.

Table 286. User Key Fields on the Opportunity Object

User Key Field Combinations
OpportunityId
IntegrationId
ExternalSystemId

Table 287 details the picklists available for the opportunity object.

Table 287. Picklists Available for the Opportunity Object

Field Name
LeadSource
Priority
Probability
ReasonWonLost
Status
Type
Year
Make
Model

For more information on the fields exposed through the Opportunity Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the opportunity object.

Partner

The partner object allows you to define and record details about partner accounts.

NOTE: To download the Partner WSDL file, you must be given access to the Partner object. If you do not have access to the Partner object, it is not available to download from the Web Services Administration screen or available to use the Web service calls. For assistance in gaining access to the Partner object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 288 details the methods called by the Partner service.

Table 288. Methods Called by Partner Service

Method	Name as Defined in Service
“Insert (Web Services v2.0)” on page 285	PartnerInsert
“Update (Web Services v2.0)” on page 294	PartnerUpdate

Fields

Table 289 details the read-only fields for the partner object.

Table 289. Read-Only Fields on the Partner Object

Field Name	Type
PartnerName	Required
AssignmentStatus	Read-only
CSN	Read-only
PartnerProgram	Read-only
ProgramPartnerType	Read-only
PartnershipBenefits	Read-only
ProgramLevel	Read-only
BeginDate	Read-only
EndDate	Read-only
Audit Fields	Read-only

Table 290 details the status key for the partner object.

Table 290. Status Key for the Partner Object

Field Name
Audit Fields
Id
ModId

Table 291 details the pickmap fields for the partner object.

Table 291. Pick Map Fields for the Partner Object

Pick Map Field	Maps To
Owner	OwnerId
ParentAccountName, ParentAccountLocation	ParentAccountId
ChannelAccountManager	ChannelAccountManagerId

Table 291. Pick Map Fields for the Partner Object

Pick Map Field	Maps To
SourceCampaignName	SourceCampaignId
Territory	TerritoryId

Table 292 provides a list of the user key combinations for the partner object.

Table 292. User Key Fields on the Partner Object

User Key Field Combinations
Id
IntegrationId
ExternalSystemId
Name & Location

Table 293 details the picklists available for the partner object.

Table 293. Picklists Available for the Partner Object

Field Name
AccountType
CallFrequency
Expertise
InfluenceType
MarketPotential
MarketSegment
PartnerOrgStatus
Priority
Region
Route
Stage
Status
IndexedPick0 to IndexedPick5

For more information on the fields exposed through the Partner Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the partner object.

Policy

The policy object allows you to define and record details about an insurance policy.

NOTE: To download the Policy WSDL file, you must be given access to the Policy object. If you do not have access to the Policy object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Policy object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 294 details the methods called by the Policy service.

Table 294. Methods Called by Policy Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	PolicyDelete
“Execute” on page 284	PolicyExecute
“Insert (Web Services v2.0)” on page 285	PolicyInsert
“QueryPage (Web Services v2.0)” on page 286	PolicyQueryPage
“Update (Web Services v2.0)” on page 294	PolicyUpdate

Fields

Table 295 details the required and read-only fields for the policy object.

Table 295. 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 296 details the status key for the policy object.

Table 296. Status Key for the Policy Object

Field Name
Audit Fields
ExternalSystemId
PolicyNumber
IntegrationId

Table 297 details the pickmap fields for the policy object.

Table 297. Pick Map Fields for the Policy Object

Pick Map Field	Maps To
ExternalId	PolicyNumber
IntegrationId	PolicyNumber

Table 298 provides a list of the user key combinations for the policy object.

Table 298. User Key Fields on the Policy Object

User Key Field Combinations
PolicyNumber
IntegrationId
ExternalSystemId

Table 299 details the picklists available for the policy object.

Table 299. Picklists Available for the Policy Object

Field Name
PolicyType
Status
SubStatus
BillingStatus

Table 299. Picklists Available for the Policy Object

Field Name
RateState
RatePlan
ReferralSource
PolicyPayMethod

For more information on the fields exposed through the Policy Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the policy object.

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.

NOTE: To download the Policy Holder WSDL file, you must be given access to the Policy Holder object. If you do not have access to the Policy Holder object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Policy Holder object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 300 details the methods called by the Policy Holder service.

Table 300. Methods Called by Policy Holder Service

Method	Name as Defined in Service
“Delete (Web Services v2.0)” on page 283	PolicyHolderDelete
“Execute” on page 284	PolicyHolderExecute
“Insert (Web Services v2.0)” on page 285	PolicyHolderInsert
“QueryPage (Web Services v2.0)” on page 286	PolicyHolderQueryPage
“Update (Web Services v2.0)” on page 294	PolicyHolderUpdate

Fields

Table 301 details the required and read-only fields for the policy holder object.

Table 301. 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 302 details the status key for the policy holder object.

Table 302. Status Key for the Policy Holder Object

Field Name
Audit Fields
ExternalSystemId
PolicyHolderName
IntegrationId

Table 303 details the pickmap fields for the policy holder object.

Table 303. Pick Map Fields for the Policy Holder Object

Pick Map Field	Maps To
ExternalId	PolicyHolderName
IntegrationId	PolicyHolderName

Table 304 provides a list of the user key combinations for the policy holder object.

Table 304. User Key Fields on the Policy Holder Object

User Key Field Combinations
PolicyHolderName
IntegrationId
ExternalSystemId

Table 305 details the picklists available for the policy holder object.

Table 305. Picklists Available for the Policy Holder Object

Field Name
InsuredType
Role

For more information on the fields exposed through the Policy Holder Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the policy holder 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.

NOTE: To download the Portfolio WSDL file, you must be given access to the Portfolio object. If you do not have access to the Portfolio object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Portfolio object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 306 details the methods called by the Portfolio service.

Table 306. Methods Called by Portfolio Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	PortfolioDelete
"Execute" on page 284	PortfolioExecute
"Insert (Web Services v2.0)" on page 285	PortfolioInsert
"QueryPage (Web Services v2.0)" on page 286	PortfolioQueryPage
"Update (Web Services v2.0)" on page 294	PortfolioUpdate

Fields

Table 307 details the read-only fields for the portfolio object.

Table 307. 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 308 details the status key for the portfolio object.

Table 308. Status Key for the Portfolio Object

Field Name
Audit Fields
ExternalSystemId
PortfolioId
IntegrationId

Table 309 details the pickmap fields for the portfolio object.

Table 309. 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 310 provides a list of the user key combinations for the portfolio object.

Table 310. User Key Fields on the Portfolio Object

User Key Field Combinations
PortfolioId
IntegrationId
ExternalSystemId

Table 311 details the picklists available for the portfolio object.

Table 311. Picklists Available for the Portfolio Object

Field Name
AccountType
Status
TermUnit

For more information on the fields exposed through the Portfolio Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the portfolio 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.

Parent Objects

Account, Campaign, and Contact

Methods Called

Table 312 details the methods called by the Product service.

Table 312. Methods Called by Product Service

Method	Name as Defined in Service
"Execute" on page 284	ProductExecute
"Insert (Web Services v2.0)" on page 285	ProductInsert
"QueryPage (Web Services v2.0)" on page 286	ProductQueryPage
"Update (Web Services v2.0)" on page 294	ProductUpdate

Fields

Table 313 details the required and read-only fields for the product object.

Table 313. Required and Read-Only Fields for the Product Object

Field Name	Type
ProductName	Required
Audit Fields	Read-only

Table 314 details the status key for the product object.

Table 314. Status Key for the Product Object

Field Name
Audit Fields
Id
IntegrationId

Table 315 details the pick map field for the product object.

Table 315. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 316 details the user keys for the product object.

Table 316. User Keys for the Product Object

Field Name
ProductId
IntegrationId
ExternalSystemId

Table 317 details the picklists available for the product object.

Table 317. Picklists Available for the Product Object

Field Name
BodyStyle
Category
Class
CurrencyCode
DoorStyle
Engine
Make
Model
PriceType
ProductType
Revision
Status
SubType
TherapeuticClass

Table 317. Picklists Available for the Product Object

Field Name
Transmission
Trim

For more information on the fields exposed through the Product Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application and generate the WSDL file for the product object.

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.

Methods Called

Table 318 details the methods called by the Service request service.

Table 318. Methods Called by Service Request Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	ServiceRequestDelete
"Execute" on page 284	ServiceRequestExecute
"Insert (Web Services v2.0)" on page 285	ServiceRequestInsert
"QueryPage (Web Services v2.0)" on page 286	ServiceRequestQueryPage
"Update (Web Services v2.0)" on page 294	ServiceRequestUpdate

Fields

Table 319 details the required and read-only fields for the service request object.

Table 319. Required and Read-Only Fields for the Service Request Object

Field Name	Type
ContactEmail	Read-only
ContactFirstName	Read-only
ContactFullName	Read-only

Table 319. Required and Read-Only Fields for the Service Request Object

Field Name	Type
ContactLastName	Read-only
Audit Fields	Read-only
LastUpdated	Read-only
ServiceRequestConcatId	Read-only

Table 320 details the status key for the service request object.

Table 320. Status Key for the Service Request Object

Field Name
Audit Fields
AccountId
ContactId
ExternalSystemId
Id
IntegrationId
LastUpdated

Table 321 details the pick map fields for the service request object.

Table 321. Pick Map Fields for the Service Request Object

Pick Map Field	Maps To
Owner	OwnerId
AccountExternalSystemId	AccountId
AssetIntegrationId	AssetId
AssetExternalSystemId	AssetId
ProductExternalSystemId	ProductId

Table 322 provides a list of the user key combinations for the service request object.

Table 322. User Key Fields on the Service Request Object

User Key Field Combinations
ServiceRequestId
IntegrationId
ExternalSystemId
SRNumber

Table 323 details the picklists available for the service request object.

Table 323. Picklists Available for the Service Request Object

Field Name
Area
Cause
Priority
Source
Status
Type

For more information on the fields exposed through the Service request Web service, go to the Web Services Administration screen 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.

Methods Called

Table 324 details the methods called by the Solution service.

Table 324. Methods Called by Solution Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	SolutionDelete
"Execute" on page 284	SolutionExecute
"Insert (Web Services v2.0)" on page 285	SolutionInsert
"QueryPage (Web Services v2.0)" on page 286	SolutionQueryPage
"Update (Web Services v2.0)" on page 294	SolutionUpdate

Fields

Table 325 details the required and read-only fields for the solution object.

Table 325. 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 326 details the status key for the solution object.

Table 326. Status Key for the Solution Object

Field Name
Audit Fields
ExternalSystemId
Id
IntegrationId
LastUpdated

Table 327 provides a list of the user key combinations for the solution object.

Table 327. User Key Fields on the Solution Object

User Key Field Combinations
SolutionId
IntegrationId
ExternalSystemId

Table 328 details the picklists available for the solution object.

Table 328. Picklists Available for the Solution Object

Field Name
Area
Cause
Priority
Source
Status
Type

For more information on the fields exposed through the Solution Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application and generate the WSDL file for the solution object.

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 329 details the methods called by the Territory service.

Table 329. Methods Called by Territory Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	TerritoryDelete
"Execute" on page 284	TerritoryExecute
"Insert (Web Services v2.0)" on page 285	TerritoryInsert
"QueryPage (Web Services v2.0)" on page 286	TerritoryQueryPage
"Update (Web Services v2.0)" on page 294	TerritoryUpdate

Fields

Table 330 details the required and read-only fields for the territory object.

Table 330. Required and Read-Only Fields for the Territory Object

Field Name	Type
TerritoryName	Required
Territory	Read-only
Audit Fields	Read-only

Table 331 details the status key for the territory object.

Table 331. Status Key for the Territory Object

Field Name
Audit Fields
ExternalSystemId
IntegrationId
TerritoryId

Table 332 details the pick map field for the territory object.

Table 332. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

The TerritoryName field is a user key for the territory object.

Table 333 details the picklist fields available for the territory object.

Table 333. Picklists Available for the Territory Object

Field Name
ParentTerritoryExternalSystemId
ParentTerritoryIntegrationId

For more information on the fields exposed through the Territory Web service, go to the Web Services Administration screen 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 the application, 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.

Methods Called

Table 334 details the methods called by the User service.

Table 334. Methods Called by User Service

Method	Name as Defined in Service
"Execute" on page 284	UserExecute
"Insert (Web Services v2.0)" on page 285	UserInsert
"QueryPage (Web Services v2.0)" on page 286	UserQueryPage
"Update (Web Services v2.0)" on page 294	UserUpdate

Fields

Table 335 details the required and read-only fields for the user object.

Table 335. 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 336 details the status key for the user object.

Table 336. Status Key for the User Object

Field Name
ModifiedById
ModifiedDate

Table 336. Status Key for the User Object

Field Name
EMailAddr
UserId
IntegrationId

Table 337 details the pick map field for the user object.

Table 337. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

Table 338 provides a list of user key combinations for the user object.

Table 338. User Key Fields on the User Object

User Key Field Combinations
UserId
ExternalSystemId
IntegrationId
EmailAddr

For more information on the fields exposed through the User Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the user 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.

NOTE: To download the Vehicle WSDL file, you must be given access to the Vehicle object. If you do not have access to the Vehicle object, it is not available to download from the Web Services Administration screen or available to use the vertical Web service calls. For assistance in gaining access to the Vehicle object, contact your Oracle CRM On Demand service provider.

Methods Called

Table 339 details the methods called by the Vehicle service.

Table 339. Methods Called by Vehicle Service

Method	Name as Defined in Service
"Delete (Web Services v2.0)" on page 283	VehicleDelete
"Execute" on page 284	VehicleExecute
"Insert (Web Services v2.0)" on page 285	VehicleInsert
"QueryPage (Web Services v2.0)" on page 286	VehicleQueryPage
"Update (Web Services v2.0)" on page 294	VehicleUpdate

Fields

Table 340 details the required and read-only fields for the vehicle object.

Table 340. 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 341 details the status key for the vehicle object.

Table 341. Status Key for the Vehicle Object

Field Name
Audit Fields
ExternalSystemId
IntegrationId
VehicleId

Table 342 details the pick map fields for the vehicle object.

Table 342. 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 343 provides a list of the user key combinations for the vehicle object.

Table 343. User Key Fields on the Vehicle Object

User Key Field Combinations
VehicleId
ExternalSystemId
IntegrationId

Table 344 details the picklists available for the vehicle object.

Table 344. Picklists Available for the Vehicle Object

Field Name
Body
Door
Engine
ExteriorColor
InteriorColor
Location
Make

Table 344. Picklists Available for the Vehicle Object

Field Name
Model
VehicleOwnedBy
Status
Transmission
Trim
UsedNew
WarrantyType
Year

For more information on the fields exposed through the Vehicle Web service, go to the Web Services Administration screen within the Oracle CRM On Demand application, and generate the WSDL file for the vehicle object.

6 Oracle CRM On Demand API Calls

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

- 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 administrative tasks through Web services.

For each of these methods, usage, arguments taken, and return values are detailed. This chapter contains the following topics:

- "Web Services v1.0 API Calls"
- "Web Services v2.0 API Calls" on page 281
- "Service API Calls" on page 295

Web Services v1.0 API Calls

The Web Services v1.0 methods are listed in Table 345. 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 345. Web Services v1.0 Methods

Method	Comments
Delete (Web Services v1.0)	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 (Web Services v1.0)	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.

Table 345. Web Services v1.0 Methods

Method	Comments
QueryPage (Web Services v1.0)	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 (Web Services v1.0)	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.

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 (Web Services v1.0)

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 346.

Table 346 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 356.

Table 346. 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

Table 346. Behavior of Delete Method on Child Objects

Parent Object	Child	Action When Parent Is Deleted
Contact	Account	None
	Activity	Delete
	Asset	None
	Campaign	None
	Interests	Delete
	Lead	Delete
	Note	Delete
	Opportunity	None
	ServiceRequest	None
Household	HouseholdTeam	None
Lead	Activity	Delete
MedEd	Invitees	None
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 347 describes the arguments taken by the Delete method.

Table 347. 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 (Web Services v1.0).

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.

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.

See "DeleteChild Sample" on page 343 for an example of XML code.

Arguments

Table 348 describes the arguments taken by the DeleteChild method.

Table 348. 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 (Web Services v1.0)

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

Table 349. Effect of Insert on Parent and Child Objects

Method	New Parent	New Child	Existing Parent	Child Exists in Oracle CRM On Demand and in Input	Child Exists in Oracle CRM On Demand but Not in Input
Insert	New parent instance	New child instance	Error	Use child instance	Not applicable

Arguments

Table 350 describes the arguments taken by the Insert method.

Table 350. 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 (Web Services v1.0).

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.

See "InsertChild Sample" on page 344 for an example of XML code.

Arguments

Table 351 describes the arguments taken by the InsertChild method.

Table 351. 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 352 outlines how the InsertOrUpdate method acts on parent and child objects to create or update an object instance.

Table 352. Effect of InsetOrUpdate on Parent and Child Objects

Method	New Parent	New Child	Existing Parent	Child Exists in Oracle CRM On Demand and in Input	Child Exists in Oracle CRM On Demand but Not in Input
InsertOrUpdate	New parent	New child	Update parent	Update child	Child is unchanged

Arguments

Table 353 describes the arguments taken by the InsertOrUpdate method.

Table 353. 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 (Web Services v1.0), Insert (Web Services v1.0)

QueryPage (Web Services v1.0)

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, 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. These queries are described in Appendix A, “Oracle CRM On Demand API Code Samples.”

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 s false.
- For a query whose record set exceeds the PageSize value, setting the StartRowNum argument 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 the 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.

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 354.

Table 354. 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`	
`AND`	
`unary operator`	
IS NULL	Used to find a match for a value that has no value
`binary operator`	
`=`	
`~=`	Denotes a case-insensitive exact search (no wildcards used)
`<`	Must be specified as &amp;lt; to ensure well-formed XML.
`<=`	Must be specified as &amp;lt;=
`>`	
`>=`	
`<>`	Must be specified as &amp;lt;>
`LIKE`	Wildcard characters are treated as such only in the context of the operator LIKE.
`~LIKE`	Denotes a case-insensitive wildcard search

Table 354. 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: \?

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. For sample expressions, see Table 383.

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, 2004 6:30am and July 28, 2004 6:45 am.

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

```
<ListOfContact>

    <Contact>

        <JobTitle />

        <ContactLastName />

        <ContactFirstName />

        <LastUpdated> (>='07/28/2004 06:30:00') AND (<='07/28/2004 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 355 describes the arguments taken by the QueryPage method.

Table 355. 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

Table 355. Arguments Taken by the QueryPage Method

Name	Description	Required	Default	I/O
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 should be included.	No		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 (Web Services v1.0)

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

Table 356. Effect of Update on Parent and Child Objects

Method	New Parent	New Child	Existing Parent	Child Exists in Oracle CRM On Demand and in Input	Child Exists in Oracle CRM On Demand but Not in Input
Update	Error	New child	Update parent	Update child	Child is removed

Arguments

Table 357 describes the arguments taken by the Update method.

Table 357. 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 records 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.

See "UpdateChild Sample" on page 345 for an example of XML code.

Arguments

Table 358 describes the arguments taken by the UpdateChild method.

Table 358. 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 359. 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 should be 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 359. Web Services v2.0 API Methods

Method	Comments
Delete (Web Services v2.0)	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 (Web Services v2.0)	Inserts new records into the Oracle CRM On Demand database.
QueryPage (Web Services v2.0)	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 (Web Services v2.0)	Updates 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.

The Echo Argument

Oracle CRM On Demand Web services using the methods Insert, Update, and Delete 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.

The ViewMode Argument

The ViewMode argument is an input argument for all of the Web Services v2.0 API methods. It specifies the level of access to records specified 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 should occur 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 (Web Services v2.0)

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.

Arguments

Table 360 describes the arguments taken by the Delete method.

Table 360. 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 283.	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 282.	No	Broadest	Input

Return Value of the Call

The status key for each of the deleted objects.

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.

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

Arguments

Table 361 describes the arguments taken by the Execute method.

Table 361. 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 283.	No	LIC	Input
ViewMode	Specifies the level of access to records specified in the method call, see "The ViewMode Argument" on page 282.	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.

Insert (Web Services v2.0)

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.

Arguments

Table 362 describes the arguments taken by the Insert method.

Table 362. 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 283.	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 282.	No	Broadest	Input

Return Value of the Call

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

QueryPage (Web Services v2.0)

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.

Using the pagesize and startownum 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.

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 291.

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 363.

Table 363. 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.

Table 363. Query Syntax for QueryPage

Syntax Type	Notes
conjunction	
OR	
AND	
unary operator	
IS NULL	Used to find a match for a value that has no value
IS NOT NULL	
EXISTS	
NOT EXISTS	
binary operator	
=	
~=	Denotes a case-insensitive exact search (no wildcards used)
<	Must be specified as &amp;lt; to ensure well-formed XML.
<=	Must be specified as &amp;lt;=
>	
>=	
<>	Must be specified as &amp;lt;>
LIKE	Wildcard characters are treated as such only in the context of the operator LIKE.
~LIKE	Denotes a case-insensitive wildcard search
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: \?

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 364.

Table 364. 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"

Table 364. Searchspec Examples

Operator	Usage of searchspec in request	Description
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
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

Table 364. Searchspec Examples

Operator	Usage of searchspec in request	Description

>
(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

<
(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

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, with a sortsequence of 1.

Examples of usage are given in Table 365.

Table 365. 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
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 Fields	<ListOfContact startrownum="0" pagesize="100" recordcountneeded="true"> <Contact searchspec=”[ContactFirstName] LIKE ‘Contact*’”> <ContactLastName sortorder="DESC" sortsequence="1"></ContactLastName> <ExternalSystemId sortorder="ASC" sortsequence="-3"></ExternalSystemId> <CreatedDate sortorder="DESC" sortsequence="-4"></CreatedDate> </Contact></ListOfContact>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with zero or more characters and sorts them by CreatedDate in descending order and then by ExternalSystemId in ascending order and then by ContactLastName 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.

Arguments

Table 366 describes the arguments taken by the QueryPage method.

Table 366. 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.	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		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 283.	No	LIC	Input
ViewMode	Specifies the level of access to records specified in the method call, see “The ViewMode Argument” on page 282.	Yes	Broadest	Input
BookId	The book ID.	No	Not applicable	Input

Table 366. Arguments Taken by the QueryPage Method

Name	Description	Required	Default	I/O
BookName	The book name. This argument is ignored if a value for BookId is supplied.	No	Not applicable	Input
IncludeSubBooks	Whether subbooks should be included.	No		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.

Update (Web Services v2.0)

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.

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. On inserting new records however, Oracle CRM On Demand checks for the required field.

Arguments

Table 367 describes the arguments taken by the Update method.

Table 367. 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 283.	No	LIC	Input

Table 367. 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 282.	Yes	Broadest	Input

Return Value of the Call

The status key for the updated objects.

Service API Calls

The Oracle CRM On Demand Web Services service methods are listed in Table 368. 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 368. Web Services On Demand Service Methods

Method Name	Web Service	Comments
"DeletedItemQueryPage" on page 296	Deleted Item	Gets information about deleted items.
"DeleteEvents" on page 300	Integration Event	Deletes events from the integration event queue.
"GetEvents" on page 301	Integration Event	Gets events from the integration event queue.
"GetPicklistValues" on page 304	Picklist	Gets lists of picklist values.
"GetMapping" on page 303	Mapping Service	Gets a list of the display names of fields for a particular record type and their associated XML tags.
"GetServerTime" on page 306	Time	Gets the server time.
"LoginHistoryQueryPage" on page 307	Login History	Gets information about user login history.

Table 368. Web Services On Demand Service Methods

Method Name	Web Service	Comments
“MergeRecords” on page 308	Merge	Merges records.
“SetPasswordAPI” on page 310	Password	Sets the passwords of users who use the application.
“SetSessionTimeZone” on page 310	Time	Sets the time zone for a session.
“UpdatePicklist” on page 311	Picklist	Updates picklist values.
“UpdateCascadingPicklists” on page 312	Picklist	Updates cascading picklist values.
“UserUsageQueryPage” on page 314	User Usage	Gets information about Web services utilization.

DeletedItemQueryPage

Returns details of deleted items.

Usage

You use the DeletedItemQueryPage method to execute a query against the list of deleted records, and returns 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 369.

NOTE: In Table 369, 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 369. 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

Table 369. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
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
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

Table 369. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
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
Custom Object 03 Attachment	OnDemand Custom Object 3 Attachment
Dealer Note	Dealer Note
Dealer Attachment	Dealer Attachment
Patient	VONDMED Patient
Patient Note	VONDMED Patient Note

Arguments

Table 370 describes the arguments taken by the DeletedItemQueryPage method.

Table 370. 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.

DeleteEvents

Deletes events from the integration event queue.

Usage

You use the DeleteEvents method of the Integration Event Web service to delete events from the integration event queue. Integration events are actions that are triggered based on meeting certain workflow criteria. The 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 a company queue on the hosted environment. The maximum number of events in the queue is set by Customer Care. Contact Customer Care to request support for the Integration Event Web Service and to specify the size of the integration queue you require.

For more information about integration events and setting up workflow criteria, refer to the Oracle CRM On Demand online help.

The DeleteEvents method deletes events from the 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.

You can delete events for all record types, or a subset of record types, depending on how you prepare the WSDL files associated with the Integration Event service, see "Downloading the Integration Event WSDL File" on page 303.

Arguments

Table 371 describes the arguments taken by the DeleteEvents method.

Table 371. Arguments Taken by the DeleteEvents Method

Name	Description	Required	Default	I/O
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.

GetEvents

Returns events from the integration event queue.

Usage

You use the GetEvents method of the Integration Event Web service to return events from the integration event queue. Integration events are actions that are triggered based on meeting certain workflow criteria. The 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 a company queue on the hosted environment. The maximum number of events in the queue is set by Customer Care. Contact Customer Care to request support for the Integration Event Web Service and to specify the size of the integration queue you require.

For more information about integration events and setting up workflow criteria, refer to the Oracle CRM On Demand online help.

If the EventCount argument is not supplied, all events are returned.

You can return events for all record types, or a subset of record types, depending on how you prepare the WSDL files associated with the Integration Event service, see "Downloading the Integration Event WSDL File" on page 303.

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 the application, as shown in Table 372.

Table 372. Object Names Returned by GetEvents Method

UI Name	GetEvents Name (object attribute)
Account Competitor	AccountCompetitor
Account Relationship	AccountRelationship
Address	CUT Address
Call Product Detail	Call ProdDetail
Call Sample Dropped	Call SampDrop
Campaign Recipient	ContactCampaign
Contact Interests	ContactInterest
Contact Relationship	ContactRelationship
Custom Object 01 Team	CustObj1 Team
Custom Object 02 Team	CustObj2 Team
Custom Object 03 Team	CustObj3 Team
Custom Object *n* Team	CustObj*n* Team

Table 372. Object Names Returned by GetEvents Method

UI Name	GetEvents Name (object attribute)
Dealer	Channel Partner
Household Team	HouseholdTeam
MedEd Event	MedEdEvent
MedEd Invitee	MedEdInvitee
Opportunity Competitor	OpportunityCompetitor
Opportunity Partner	OpportunityPartner
Portfolio Team	PortfolioTeam
Portfolio Owner	PortfolioOwners
Vehicle Financial Information	Vehicle FinInfo
Vehicle Sales History	Vehicle SalesHist
Vehicle Service History	Vehicle ServHist

Arguments

Table 373 describes the arguments taken by the GetEvents method.

Table 373. 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
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.

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.

Downloading the Integration Event WSDL File

You can use the methods of the Integration Events service to track changes for particular record types or for all record types, depending on how you prepare the WSDL.

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.

   The file downloaded is the integrationevent.zip file. This zip file contains the integration event WSDL file and the XSD schema files of all supported record types such as account, contact, and so on. The integration event WSDL file imports the XSD files for each record type.
3. Unzip the integrationevent.zip file to the required location.

   You must extract the integration event WSDL file and the XSD schema files to the same folder.
4. Add the integration event WSDL to your development environment.

Downloading Schema Files for Record Types Containing Custom Fields

If you have record types that have custom or renamed fields, you cannot use the XSD files contained in the integrationevent.zip file for tracking these fields. Instead you must download an XSD file using the Download Custom Schema button in the Web Services Administration page in the application. For more information, refer to online help for Oracle CRM On Demand.

GetMapping

Returns the display names and XML tags of the fields of a record type or one of its child components.

Usage

You use the GetMapping method to return the display names of all the fields in a particular record type. It also returns the XML tags for each field. It therefore displays the mappings between the display names of fields and their XML tags.

This method can be used on all record types and on all of their child components.

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

Arguments

Table 374 describes the arguments taken by the GetMapping method.

Table 374. 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.
- **ElementName**. The XML element name for the field.
- **DataType**. The field type of the field, for example, Check box, Picklist, and so on.

GetPicklistValues

Gets picklist values from Oracle CRM On Demand.

GetPicklistValues is supported for the following record types:

- Account
- Activity
- Asset
- Campaign
- Category (Product Category)
- Contact
- Custom Object 01 to 03
- Dealer
- Household

- Lead
- MedEdEvent
- Opportunity
- Portfolio
- Product
- RevenueAsset (Opportunity Product)
- Service Request
- Solution
- User
- Vehicle

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. This is illustrated in "Sample of GetPicklistValues for a Cascading Picklist" on page 349.

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. For an illustration of this, see "Sample of GetPicklistValues for a Cascading Picklist" on page 349.

If a picklist is not cascading, the following elements are returned empty:

- ParentFieldName
- ParentDisplayValue
- ParentCode

For an illustration of this, see "Sample of GetPicklistValues for a Regular Picklist" on page 351.

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 375 describes the arguments taken by the GetPicklistValues method.

Table 375. 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).

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.

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 376 describes the arguments taken by the LoginHistoryQueryPage method.

Table 376. 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

Table 376. Arguments Taken by the LoginHistoryQueryPage Method

Name	Description	Required	Default	I/O
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.

MergeRecords

Merges records for certain record types.

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*. 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.

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.

Table 377 describes the arguments taken by MergeRecords.

Table 377. Arguments taken by MergeRecords

Field Name		Required	Default	I/O
PrimaryId	The ID of the primary record.	Yes	Not applicable	Input
PrimaryExternalSystemId	The externalsystemID of the primary record.	Yes	Not applicable	Input
DuplicateId	The ID of the duplicate record.	No	Not applicable	Input
DuplicateExternalSystemId	The externalsystemID of the duplicate record.	No	Not applicable	Input
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.

SetPasswordAPI

Allows the system administrator to set the passwords of users who use the application.

Usage

You use the SetPasswordAPI 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 SetPasswordAPI, they can call Customer Care to have the functionality enabled.

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 378 describes the arguments taken by SetPasswordAPI.

Allows the system administrator to set the passwords of users who use the application.

Table 378. Arguments Taken by SetPasswordAPI

Field Name	Description	Required	Default	I/O
UserId	The user ID	Yes	Not applicable	Input/Output
EmailAddr	The user’s email address	No	Not applicable	Input/Output
UserIntegrationID	The integration Id for the user.	No	Not applicable	Input/Output
IntegrationId	The integration Id	No	Not applicable	Input/Output
Password	The password for the user.	Yes	Not applicable	Input/Output

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 379 describes the arguments taken by the SetSessionTimeZone method.

Table 379. 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 current server time converted to the specified time zone.	Not applicable	Not applicable	Output

Return Value of the Call

The current server time.

UpdatePicklist

Updates picklist values in Oracle CRM On Demand.

Usage

You use the UpdatePicklist method to update lists of values, typically in a language-dependent manner. UpdatePicklist is supported for the same record types as supported by the GetPicklistValues method, see "GetPicklistValues" on page 304.

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.
- **MarkTranslate.** Whether the value is marked 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 380 describes the arguments taken by the UpdatePicklist method.

Table 380. 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
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.

UpdateCascadingPicklists

Updates cascading picklist values in Oracle CRM On Demand.

Usage

You use the UpdateCascadingPicklists method to update cascading picklists, typically in a language-dependent manner.

UpdateCascadingPicklists is supported for the same record types as supported by the GetPicklistValues method, see "GetPicklistValues" on page 304.

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 381 describes the arguments taken by the UpdateCascadingPicklists method.

Table 381. 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

Table 381. Arguments Taken by the UpdateCascadingPicklists Method

Name	Description	Required	Default	I/O
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.

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 382 describes the arguments taken by the UserUsageQueryPage method.

Table 382. 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
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.

A Oracle CRM On Demand API Code Samples

This appendix contains samples of code for logging in and out of a Web services session and XML code for using the methods of the Web services API. It contains the following topics:

- "Code Samples for Logging In and Logging Out" on page 317
- "Code Samples for Logging In Using Single Sign-On" on page 326
- "XML Code Samples for API Calls" on page 333

Code Samples for Logging In and Logging Out

The following code samples illustrate how to log in and log out using Visual Basic (VB), Java, and C# .NET code. These samples are specific to Oracle CRM On Demand Web Services. For clarity and simplicity, the following samples do not include error detection and handling. The code samples are detailed in the following topics:

- "Visual Basic Code Sample" on page 317
- "Java Code Sample" on page 320
- "C# Code Sample" on page 323

Visual Basic Code Sample

The following sample is compatible with Visual Studio, Visual Basic 6.0.

```
' Declarations used to perform http internet communications from Visual Basic.

Declare Function InternetOpen Lib "WININET.DLL" Alias "InternetOpenA" ( _

   ByVal lpszAgent As String, _

   ByVal dwAccessType As Long, _

   ByVal lpszProxyName As String, _

   ByVal lpszProxyBypass As String, _

   ByVal dwFlags As Long) As Long

Declare Function InternetConnect Lib "WININET.DLL" Alias "InternetConnectA" ( _

   ByVal hInternetSession As Long, _

   ByVal lpszServerName As String, _
   ByVal nServerPort As Integer, _

   ByVal lpszUserName As String, _

   ByVal lpszPassword As String, _

   ByVal dwService As Long, _

   ByVal dwFlags As Long, _

   ByVal dwContext As Long) As Long

Declare Function HttpSendRequest Lib "WININET.DLL" Alias "HttpSendRequestA" ( _

   ByVal hHttpRequest As Long, _

   ByVal lpszHeaders As String, _

   ByVal dwHeadersLength As Long, _

   ByVal lpOptional As String, _

   ByVal dwOptionalLength As Long) As Integer

Declare Function InternetReadFile Lib "WININET.DLL" ( _

   ByVal hFile As Long, _

   ByRef lpBuffer As Any, _

   ByVal dwNumberOfBytesToRead As Long, _

   ByRef lpNumberOfBytesRead As Long) As Integer

Declare Function InternetCloseHandle Lib "WININET.DLL" ( _
   ByVal hInternet As Long) As Boolean

' Adds one or more HTTP request headers to the HTTP request handle.

Declare Function HttpAddRequestHeaders Lib "WININET.DLL" Alias HttpAddRequestHeadersA"
( _

   ByVal hHttpRequest As Long, _

   ByVal sHeaders As String, _

   ByVal lHeadersLength As Long, _

   ByVal lModifiers As Long) As Integer

Declare Function HttpQueryInfo Lib "WININET.DLL" Alias "HttpQueryInfoA" ( _

   ByVal hHttpRequest As Long, _

   ByVal lInfoLevel As Long, _

   ByRef sBuffer As Any, _

   ByRef lBufferLength As Long, _
   ByRef lIndex As Long) As Integer

' Define a function or sub to contain the login steps as defined here.

       Dim netHeaders As String * 1024

       Dim headerSize as long

' Open the internet connection.

   m_hInternet = InternetOpen("ApplicationName", 0, vbNullString, vbNullString, 0)

   m_hConnect = InternetConnect(m_hInternet, server, 443, sProxyUser, sProxyPwd, 3, 0,
   0)

   m_hRequest = HttpOpenRequest(m_hConnect, "GET", "/Services/
   Integration?command=login", "HTTP/1.1", vbNullString, vbNullString, &H84A83000, 0)

   sHeaders = "Accept-Language: en" & vbCrLf & _
      "Connection: Keep-Alive" & vbCrLf & _

      "UserName: email@address.com" & vbCrLf & _

      "Password: userpassword"

   result = HttpAddRequestHeaders(m_hRequest, sHeaders, Len(sHeaders), &HA0000000)

   ' Add headers

   result = HttpSendRequest(m_hRequest, vbNullString, 0, vbNullString, 0)

   headerSize = Len(netHeaders)

   result = HttpQueryInfo(m_hRequest, &H16, ByVal netHeaders, headerSize, 0)

   ' Now parse netHeaders for header named JSESSIONID.

   ' This will be the value to use for the rest of the session.

' Then to Logoff, do the following:

   m_hRequest = HttpOpenRequest(m_hConnect, "GET", ")/Services/
   Integration?command=logoff", "HTTP/1.1", vbNullString, vbNullString, &H84A83000, 0)

   sHeaders = "Accept-Language: en" & vbCrLf & _

      "Connection: Keep-Alive" & vbCrLf & _

      "Cookie: JSESSIONID=abc123:-1"

   result = HttpAddRequestHeaders(m_hRequest, sHeaders, Len(sHeaders), &HA0000000)

   ' Add headers

   result = HttpSendRequest(m_hRequest, vbNullString, 0, vbNullString, 0)
```

Java Code Sample

The following is a code sample for logging in and logging out using Java:

```
/*

   * log on to a web services session at the passed in service provider location using
   * the passed in credentials. return a session id string that can be used in later

   * communication with the service provider in the event of a successful logon

   * @param wsLocation - the location of the web services provider

   * @param userName - On Demand user name (email address) of the user we are logging
   * in as

   * @param password - password that corresponds to the userName parameter

   * @return FAIL if the logon failed, session id string otherwise

   *

 */

private static String logon(String wsLocation, String userName, String password)

{

   String sessionString = FAIL;

   try

   {

      // create an HTTPS connection to the On Demand webservices

      URL wsURL = new URL(wsLocation + "?command=login");

      HttpURLConnection wsConnection = (HttpURLConnection)wsURL.openConnection();


      // we don't want any caching to occur

      wsConnection.setUseCaches(false);


      // we want to send data to the server

      // wsConnection.setDoOutput(true);


      // set some http headers to indicate the username and password we are using to
      logon

      wsConnection.setRequestProperty("UserName", userName);
      wsConnection.setRequestProperty("Password", password);
      wsConnection.setRequestMethod("GET");

      // see if we got a successful response
      if (wsConnection.getResponseCode() == HttpURLConnection.HTTP_OK)
   {
      // get the session id from the cookie setting
      sessionString = getCookieFromHeaders(wsConnection);
      setSessionIdFromCookie(sessionString);
   }
   }
   catch (Exception e)
   {
      System.out.println("Logon Exception generated :: " + e);
   }
   return sessionString;
}

/*
   * log off an existing web services session, using the sessionCookie information
   * to indicate to the server which session we are logging off of
   * @param wsLocation - location of web services provider
   * @param sessCookie - cookie string that indicates our sessionId with the WS provider
   *
*/
private static void logoff(String wsLocation, String sessionCookie)
{
     try
     {
         // create an HTTPS connection to the On Demand webservices
```

```
          URL wsURL = new URL(wsLocation + "?command=logoff");
          HttpURLConnection wsConnection = (HttpURLConnection)wsURL.openConnection();

       // we don't want any caching to occur
       wsConnection.setUseCaches(false);

       // let it know which session we're logging off of
       wsConnection.setRequestProperty("Cookie", "JSESSIONID=" + sessionCookie);
       wsConnection.setRequestMethod("GET");

       // see if we got a successful response
       if (wsConnection.getResponseCode() == HttpURLConnection.HTTP_OK)
    {
       // if you care that a logoff was successful, do that code here
       // showResponseHttpHeaders(wsConnection);
    }
    }
    catch (Exception e)
    {
       System.out.println("Logoff Exception generated :: " + e);
    }
}

/*
    * given a successful logon response, extract the session cookie information
    * from the response HTTP headers
    *
    * @param wsConnection successfully connected connection to On Demand web services
    * @return the session cookie string from the On Demand WS session or FAIL if not
    found*
   */
```

```
private static String getCookieFromHeaders(HttpURLConnection wsConnection)
   {
      // debug code - display all the returned headers
      String headerName;
      String headerValue = FAIL;
      for (int i=0; ; i++)
   {
   headerName = wsConnection.getHeaderFieldKey(i);
   if (headerName != null && headerName.equals("Set-Cookie"))
      {
         // found the Set-Cookie header (code assumes only one cookie is being set)
         headerValue = wsConnection.getHeaderField(i);
         break;
      }
   }
   // return the header value (FAIL string for not found)
   return headerValue;
}
```

C# Code Sample

The following is a code sample for logging in and logging out using C# (C Sharp):

```
using System;
using System.Net;
using System.IO;

namespace WebServiceHandler
{

   public class ManageSession
```

```
{

public static string SessionID = "";

public static String Login(String loginUrlString, String userName, String password,
StringBuilder output)
{
   try
   {
      // create a http request and set the headers for authentication
      HttpWebRequest myRequest = (HttpWebRequest)WebRequest.Create(loginUrlString);
      HttpWebResponse myResponse;

      myRequest.Method = "GET";
      myRequest.Headers["UserName"] = userName;
      myRequest.Headers["Password"] = password;

      // Return the response.
      myResponse = (HttpWebResponse)myRequest.GetResponse();
      Stream sr = myResponse.GetResponseStream();

      // retrieve session id
      char[] sep = { ';' };

      String[] headers = myResponse.Headers["Set-Cookie"].Split(sep);
      for (int i=0; i <= headers.Length-1; i++)
      {
      if (headers[i].StartsWith("JSESSIONID"))
      {
      sep[0] = '=';
```

```
        SessionID = headers[i].Split(sep)[1];
        break;
        }
    }
    sr.Close();
    myResponse.Close();
}
catch (WebException webException)
{
}
catch (Exception e)
{
}
return SessionID;
}

public static void Logoff()
    {
        String logoffUrlString = serverName + "/Services/Integration?command=logoff";
        HttpWebRequest req = (HttpWebRequest) WebRequest.Create(logoffUrlString);
        req.Headers["Cookie: JSESSIONID"] = SessionID;

        // make the HTTP call
        HttpWebResponse resp = (HttpWebResponse) req.GetResponse();
        if (resp.StatusCode != System.Net.HttpStatusCode.OK)
        {
    }
}
}
}
```

Code Samples for Logging In Using Single Sign-On

The following code samples illustrate how to log in using single sign-on using Java, and C# .NET code. These samples are specific to Oracle CRM On Demand Web Services. The code samples are detailed in the following topics:

- "Java Code Sample" on page 326
- "C# Code Sample" on page 331

Java Code Sample

The following is a code sample for single sign-on using Java:

```
//Declare variables for sessionId and servername

private static String mstrSessionId="";

private static String mstrServer="";


public static String logonSSO(String ssoid)

{

   String sessionString = "FAIL";

   String pftokenCookie = "";

   String sessionCookie = "";

   String strSSOITSURL="";

   String newlocation="";

   boolean blnGotSession = false;


   try

   {


      // Retrieve the ITS url from OnDemand

      //If you want to use a different ITS url, this step can be skipped

      // create an HTTPS connection to OnDemand
```

```
URL wsURL = new URL(mstrServer + "/Services/Integration?command=ssoitsurl&ssoid="
+ ssoid);

HttpURLConnection wsConnection = (HttpURLConnection) wsURL.openConnection();


//Turn off caching

wsConnection.setUseCaches(false);

wsConnection.setDoOutput(true);

wsConnection.setRequestMethod("GET");


// If response is successful retrieve ssoitsurl

if (wsConnection.getResponseCode() == HttpURLConnection.HTTP_OK)

{


   //retrieve ssoitsurl string from response

   strSSOITSURL = wsConnection.getHeaderField("X-SsoItsUrl");


   //Open a connection to the ITS URL

   wsURL = new URL(strSSOITSURL);

   wsConnection = (HttpURLConnection) wsURL.openConnection();


   //Turn off caching

   wsConnection.setUseCaches(false);


   //The ITS URL will redirect and pass a PTFToken

   //Turn off redirects and redirect manually so that

   //the token can be passed along to the url which will return the sessionId

   wsConnection.setInstanceFollowRedirects(false);

   wsConnection.setRequestMethod("GET");


   while(!blnGotSession)
```

```
{
        //Call CookieFromHeaders function to return cookie
        //Retrieve the PFTOKEN20 cookie so we can pass it in the redirect
        pftokenCookie = getCookieFromHeaders(wsConnection,"PFTOKEN20");

        //check to see if we have a jsession cookie so we can stop redirecting
        sessionCookie = getCookieFromHeaders(wsConnection,"JSESSIONID");

        //Check to see if we have the sessionid and stop redirecting
        if(sessionCookie != "FAIL")
    {
        blnGotSession = true;
    }
        else
    {
        //Get the URL for the next Re Direct
        String newurl = wsConnection.getHeaderField("Location");
        wsURL = new URL(newurl);
        wsConnection = (HttpURLConnection) wsURL.openConnection();
        wsConnection.setInstanceFollowRedirects(false);
        wsConnection.setUseCaches(false);

        if(pftokenCookie != "FAIL")
        {
        wsConnection.setRequestProperty("Cookie", pftokenCookie);
        }

        wsConnection.setRequestMethod("GET");
    }
}
```

```
            if (blnGotSession)

         {

            // Retrieve the session id from the cookie setting

            setSessionIdFromCookie(sessionCookie);

            return mstrSessionId;


         }

      }

      catch (Exception e)

      {

      System.out.println("Logon Exception generated :: " + e);

      }

      return mstrSessionId;

   }




   //Function which returns cookie from headers

   //If no cookie is returned, return FAIL

   private static String getCookieFromHeaders(HttpURLConnection wsConnection,String 
   pstrCookieName)

   {



      String headerName;

      String headerValue = "FAIL";

      Map m = wsConnection.getHeaderFields();


      for (Iterator it = m.keySet().iterator();it.hasNext();)

      {

         String headerFieldKey = (String)it.next();
```

```
        Object headerFieldValue = (Object) m.get(headerFieldKey);

    }


    for (int i = 0;i <= m.size() ; i++)

    {


        headerName = wsConnection.getHeaderFieldKey(i);


        if (headerName != null && headerName.equals("Set-Cookie") &&
        wsConnection.getHeaderField(i).indexOf(pstrCookieName)>=0 )

        {

            headerValue = wsConnection.getHeaderField(i);

            break;

        }

    }


    // return the header value (FAIL string for not found)

    return headerValue;

}


//Strip JSESSIONID from cookie

private static void setSessionIdFromCookie(String sessionString)

{

    int jsessionIdx=0;


    jsessionIdx = sessionString.indexOf("JSESSIONID=");

    if (jsessionIdx >=0)

    {

        //remove anything before the JSESSIONID=

        sessionString = sessionString.substring(jsessionIdx);
        // 11 is the length of JSESSIONID=
        setSessionId(sessionString.substring(11, sessionString.indexOf(";")));
    }
}

//Set SessionId
public static void setSessionId(String sessionId)
{
    mstrSessionId = sessionId;
}
```

C# Code Sample

The following is a code sample for single sign-on using C# (C Sharp):

```
public void SSOEstablish(string server, string ssoid)
{
    DateTime dtStart = DateTime.Now;
    string strSSOITSURL="";
    string Servername = server;
    string SSOID = ssoid;

    try
    {
        //check for existing session
        if(sessionId != null)
        {
        this.Destroy();
        }
```

```
// create a container for an HTTP request

HttpWebRequest req = (HttpWebRequest) WebRequest.Create(Servername + "/Services/
Integration?command=ssoitsurl&ssoid=" + SSOID);


// make the HTTP call

HttpWebResponse resp=null;

try

{

   resp = (HttpWebResponse) req.GetResponse();

}

catch(Exception excep)

{

   throw(excep);

}

//if call is succesful, retrieve the SSOITSURL

if (resp.StatusCode == System.Net.HttpStatusCode.OK)

{

   strSSOITSURL= resp.Headers["X-SsoItsUrl"];

}

//make a call to ssoitsurl

req = (HttpWebRequest) WebRequest.Create(strSSOITSURL);


// cookie container is added to the request

// retrieve the cookie from the response.

req.CookieContainer = new CookieContainer();

try

{

   resp = (HttpWebResponse) req.GetResponse();

}

catch(Exception excep)
```

```
        {
            throw(excep);
        }
        if (resp.StatusCode == System.Net.HttpStatusCode.OK)
        {
            //store cookie
            cookie = resp.Cookies["JSESSIONID"];
            if (cookie == null)
            {
            throw new Exception("No JSESSIONID cookie found in log-in response!");
            }
            //obtain sessionID
            sessionId = cookie.Value;
        }
    }
    catch(Exception excep)
    {
    throw(excep);
    }

}
```

XML Code Samples for API Calls

This appendix contains samples of XML code entered in and returned from some of the Oracle CRM On Demand Web services methods.

- "QueryPage Method: Sample Expressions" on page 334
- "QueryPage Method: Query by Template Samples" on page 336
- "QueryPage Method: Query by Children Samples" on page 338
- "Child Node Method Samples" on page 343
- "MergeRecords Method Sample" on page 346
- "DeleteEvents Method Sample" on page 347

- "GetEvents Method Sample" on page 347
- "GetPicklistValues Method Samples" on page 349

QueryPage Method: Sample Expressions

This topic contains examples describing the use of various query values in the QueryPage method and the resulting record set that are returned. For more information about the QueryPage method, see "QueryPage (Web Services v1.0)" on page 273.

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

Table 383. 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.

Table 383. Returned Record Sets

Field Value	Returned Record Set	Comments
= '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
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

Table 383. Returned Record Sets

Field Value	Returned Record Set	Comments
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.
IS NULL	(empty field) ((> 'abc*') AND (< 'abcd'))	None
OR (~= 'abc*d')	abc*d aBc*D abc*d	None
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.

QueryPage Method: Query by Template Samples

This topic contains samples of XML code that is relevant to the QueryPage method when using the Query by Template type of query. For more information about the QueryPage method, see "QueryPage (Web Services v1.0)" on page 273.

The Oracle CRM On Demand Export API has *Query by Template* semantics. This means that only components and fields that are present in the input parameter are present in the output parameter.

Query by Template Example 1

The following is an example of an input integration object instance:

```
<ListOfAccount>

    <Account>

        <LastUpdated>&gt;= '10/13/2003 03:25:32'</LastUpdated>

        <Name />
```

```
      <Location>IS NULL</Location>
   </Account
</ListOfAccount>
```

The search specification applied to the Account BusComp is as follows:

```
[LastUpdated] >= '10/13/2003 03:25:32' AND [Location] IS NULL.
```

The following shows an example of the XML returned:

```
<ListOfAccount>
   <Account>
      <LastUpdated>10/14/2003 04:25:32</LastUpdated>
      <Name>IBM</Name>
      <Location></Location>
   </Account>
   <Account>
      <LastUpdated>10/13/2003 03:25:36 PM</LastUpdated>
      <Name>Siebel</Name>
      <Location></Location>
   </Account>
</ListOfAccount>
```

Query by Template Example 2

The following is an example of an input integration object instance:

```
<ListOfAccount>
   <Account>
      <LastUpdated/>
      <Name>LIKE 'Sie*'<Name/>
      <Location>= 'San Mateo'</Location>
   </Account>
</ListOfAccount>
```

The application applies the search specification as follows:

```
[Name] LIKE 'Sie*' AND [Location] = 'San Mateo'
```

The following is an example of the XML returned:

```
<ListOfAccount>

   <Account>

      <LastUpdated>10/19/2003 09:22:33 AM</LastUpdated>

      <Name>Siebel Systems, Inc.</Name>

      <Location>San Mateo</Location>

   </Account>

   <Account>

      <LastUpdated>8/22/2003 03:25:36 PM</LastUpdated>

      <Name>Siemens</Name>

      <Location>San Mateo</Location>

   </Account>

</ListOfAccount>
```

QueryPage Method: Query by Children Samples

This topic contains samples of XML code that is relevant to the QueryPage method when using the Query by Children type of query. This is applicable for Web Services v1.0 only. For more information about the QueryPage method, see "QueryPage (Web Services v1.0)" on page 273.

The examples in this topic illustrate the semantics of queries that include search specifications on child components.

Query by Children Example 1

This example illustrates data synchronization through periodic exports, based on the time of the last update:

```
<ListOfAccount>

  <Account>

    <Name></Name>

    <Location></Location>

    <LastUpdated>(&gt; '05/12/02 10:00:03 PM') AND (&lt;= '05/18/02 10:02:22 PM')</
LastUpdated>

    <ListOfContact>

      <Contact>
```

```
          FirstName><FirstName>

          <LastName></Lastname>

          <LastUpdated>(&gt; '05/12/02 10:00:03 PM') AND (&lt;= '05/18/02 10:02:22
PM')</LastUpdated>

       </Contact>

     </ListOfContact>

     <ListOfBusinessAddress>

       <BusinessAddress>

          <Address/>

          <City/><State/>

          <ZipCode/>

          <LastUpdated>(&gt; '05/12/02 10:00:03 PM') AND (&lt;= '05/18/02 10:02:22
PM')</LastUpdated>

       <BusinessAddress>

     </ListOfBusinessAddress>

  </Account>

</ListOfAccount>
```

Using a formal search specification language, the application interprets this query as follows:

```
(([Account_LastUpdated] > '05/12/02 10:00:03 PM')

  AND

  ([Account_LastUpdated] <= '05/18/02 10:02:22 PM')

)

OR

EXISTS

(

  ([Contact_LastUpdated] >= '05/12/02 10:00:03 PM')

  AND

  ([Contact_LastUpdated] <= '05/18/02 10:02:22 PM')

)

OR

EXISTS
```

```
(
  ([BusinessAddress_LastUpdated] > '05/12/02 10:00:03 PM')
  AND
  ([BusinessAddress_LastUpdated] <= '05/18/02 10:02:22 PM')
)
```

This specifies to find all accounts that have met one of the following conditions between 05/12/02 10:00:03 PM and 05/18/02 10:02:22 PM in the current user's time zone that have:

- Been updated
- Been associated with at least one new contact
- Changed or added at least one business address

Query by Children Example 2

This example shows how to query to find all child opportunities associated with a contact:

```
<ListOfOpportunity>
  <Opportunity>
     <Description/>
     <Revenue></Revenue>
     <ListOfContact>
        <Contact>
           <Id>(= '12-12345') OR (= '12-54321')</Id>
           <FirstName></FirstName>
           <LastName></Lastname>
        </Contact>
     </ListOfContact>
  </Opportunity>
</ListOfOpportunity>
```

Using a formal search specification language, the application interprets this query as follows:

```
EXISTS( [Contact_Id]='12-12345' OR [Contact_Id]='12-54321' )
```

This example specifies to find all opportunities that are associated either with a contact whose Id is 12-12345, or with a contact whose Id is 12-54321.

Query by Children Example 3

This example illustrates the interpretation of peer inter-component and intra-component expressions, and the fact that the default operator for text fields is LIKE:

```
<ListOfAccount>
  <Account>
     <Name></Name>
     <Location></Location>
     <LastUpdated>&gt; 05/12/02 10:00:03 PM</LastUpdated>
     <ListOfContact>
        <Contact>
          <FirstName>= 'Sanjin'<FirstName>
          <LastName>= 'Tulac'</Lastname>
          <MiddleName />
        </Contact>
        <Contact>
          <FirstName>LIKE 'Alex*'<FirstName>
          <LastName>LIKE 'Warsha*'</Lastname>
        </Contact>
     </ListOfContact>
     <ListOfOpportunity>
        <Opportunity>
          <Description/>
          <Revenue>&gt;= '10000'</Revenue>
        </Opportunity>
     </ListOfOpportunity>
  </Account>
</ListOfAccount>
```

The application interprets this input to the QueryPage method as the following search specification:

```
[Account_LastUpdated] >= '05/12/02 10:00:03 PM'
OR
```

```
EXISTS
(
  (
  ([Contact_FirstName]='Sanjin') AND ([Contact_LastName]='Tulac')
  )
  OR
  (
  ([Contact_FirstName]LIKE 'Alex*') AND [Contact_LastName]LIKE 'Warsha*')
  )
)
OR
EXISTS
(
  [Opportunity_Revenue] >= '10000'
)
```

This example specifies to find all accounts that:

- Have been updated since 05/12/02 10:00:03 PM in current user's time zone
- Are associated with a contact whose first name is Sanjin and whose last name is Tulac
- Are associated with a contact whose first name starts with Alex and whose last name starts with Warsha
- Have an associated opportunity whose revenue is estimated at more than 10000 units of currency

Query by Children Example 4

This example illustrates the use of the UseChildAnd parameter to retrieve a list of modified activities for a user:

```
<UseChildAnd>true</UseChildAnd>
<ListOfActivity>
  <Activity>
     <Subject/>
     <Description/>
     <Location/>
```

```
    <ModifiedDate>&gt; '05/12/02 10:00:03 PM'</ModifiedDate>

    <ListOfUser>

       <User>

          <UserId>='12-12345'</UserId>

       </User>

    </ListOfUser>

    </Activity>

</ListOfActivity>
```

Using a formal search specification language, the application interprets this query as follows:

```
([Activity_ModifiedDate] > '05/12/02 10:00:03 PM')

AND

EXISTS

 ([User_Id]='12-12345')
```

This example tells the application to find all activities that have met all of the following conditions after 05/12/02 10:00:03 PM in the current user's time zone that:

- Have been updated
- Are associated with a user whose Id is 12-12345

Child Node Method Samples

This topic contains samples for the child-node methods InsertChild, UpdateChild, and DeleteChild of the Web Services v1.0 API.

DeleteChild Sample

This sample shows a request to delete a contact child record of an account. The parent account is identified by the AccountId field with the value 1-DOLTV, and contact child record is identified using the ContactId field with the value 1-EA7P7.

```
<?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:xsd="http://
www.w3.org/2001/XMLSchema">

   <soap:Body>

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

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

```
            <Account>

               <AccountId>1-DOLTV</AccountId>

               <ListOfContact>

                  <Contact>

                     <ContactId>1-EA7P7</ContactId>

                  </Contact>

               </ListOfContact>

            </Account>

         </ListOfAccount>

      </AccountWS_AccountDeleteChild_Input>

   </soap:Body>

</soap:Envelope>
```

InsertChild Sample

This sample shows a request to insert a contact child record of an account. The parent account is identified by the AccountId field with the value 1-DOLTV,and the contact child record is identified using the ContactId field with the value 1-EA7P7. Values for the ContactFirstName, ContactLastName, and ContactRole fields are provided.

```
<?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:xsd="http://
www.w3.org/2001/XMLSchema">

   <soap:Body>

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

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

            <Account>

               <AccountId>1-DOLTV</AccountId>

               <ListOfContact>

                  <Contact>

                    <ContactId>1-EA7P7</ContactId>

                     <ContactFirstName>Freddy15</ContactFirstName>
                     <ContactLastName>Quimby15</ContactLastName>
                     <ContactRole>User</ContactRole>
                  </Contact>
               </ListOfContact>
            </Account>
         </ListOfAccount>
      </AccountWS_AccountInsertChild_Input>
   </soap:Body>
</soap:Envelope>
```

UpdateChild Sample

This sample shows a request to update a contact child record of an account. The parent account is identified by the AccountId field with the value 1-DOLTV, and the contact child record is identified using the ContactId field with the value 1-EA7P7. Values for updating the ContactFirstName, ContactLastName, and ContactRole fields are provided.

```
<?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:xsd="http://
www.w3.org/2001/XMLSchema">
   <soap:Body>
  <AccountWS_AccountUpdateChild_Input xmlns="urn:crmondemand/ws/account/10/2004">
         <ListOfAccount xmlns="urn:/crmondemand/xml/account">
            <Account>
               <AccountId>1-DOLTV</AccountId>
               <ListOfContact>
                  <Contact>
                     <ContactId>1-EA7P7</ContactId>
                     <ContactFirstName>Freddy22_updt</ContactFirstName>
                     <ContactLastName>Quimby22_updt</ContactLastName>
                     <ContactRole>User</ContactRole>
                  </Contact>
               </ListOfContact>
            </Account>

         </ListOfAccount>

      </AccountWS_AccountUpdateChild_Input>

</soap:Body>

</soap:Envelope>
```

MergeRecords Method Sample

This topic contains a sample SOAP request and SOAP response for the MergeRecords method.

The following is the SOAP request:

```
<?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>

     <MergeRecordsWS_MergeRecords_Input xmlns="urn:crmondemand/ws/mergerecords/">

       <PrimaryId>1-E5C55</PrimaryId>

       <DuplicateId>1-E5C4Z</DuplicateId>

       <PrimaryExternalSystemId />

       <DuplicateExternalSystemId />

       <RecordType>Contact</RecordType>

     </MergeRecordsWS_MergeRecords_Input>

  </soap:Body>

</soap:Envelope>
```

The following is the SOAP response:

```
<?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:MergeRecordsWS_MergeRecords_Output xmlns:ns="urn:crmondemand/ws/
mergerecords/">

       <ns:DeletedRecordExternalSystemId />
```

```
        <ns:DeletedRecordId>1-E5C4Z</ns:DeletedRecordId>

        <ns:MergedRecordExternalSystemId />

        <ns:MergedRecordId>1-E5C55</ns:MergedRecordId>

      </ns:MergeRecordsWS_MergeRecords_Output>

    </SOAP-ENV:Body>

  </SOAP-ENV:Envelope>
```

DeleteEvents Method Sample

This topic contains a sample SOAP request for the DeleteEvents method.

```
<?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>

  <IntegrationEventWS_DeleteEvents_Input xmlns="urn:crmondemand/ws/
integrationevent/">

       <LastEventId>20061128121620_Account_Account_1-FLSDF_0_i.xml</LastEventId>

     </IntegrationEventWS_DeleteEvents_Input>

  </soap:Body>

  </soap:Envelope>
```

GetEvents Method Sample

This topic contains a sample SOAP request and SOAP response for the GetEvents method.

The following is the SOAP request:

```
<?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>

<IntegrationEventWS_GetEvents_Input xmlns="urn:crmondemand/ws/integrationevent/">

       <EventCount/>

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

The following is the SOAP response:

```
<?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:IntegrationEventWS_GetEvents_Output xmlns:ns="urn:crmondemand/ws/
integrationevent/">
      <ListOfEvent xmlns="urn:/crmondemand/xml/integrationevent">
        <Event name="TerritoryNew">
          <SiebelMessage>
            <ListOfTerritory xmlns="urn:/crmondemand/xml/territory">
              <Territory operation="insert">
                <TerritoryId>1-GF2YV</TerritoryId>
                <CurrencyCode>USD</CurrencyCode>
                <TerritoryExternalSystemId />
                <ModifiedDate>03/16/2007 18:15:09</ModifiedDate>
                <TerritoryIntegrationId>1-GF2YV</TerritoryIntegrationId>
                <ModId>0</ModId>
                <CreatedDate>03/16/2007 18:15:09</CreatedDate>
                <CurrentQuota>12345</CurrentQuota>
                <Description>Description</Description>
                <TerritoryName>myTerritory</TerritoryName>
                <ModifiedById>1-D5XDS</ModifiedById>
                <CreatedById>1-D5XDS</CreatedById>
              </Territory>
            </ListOfTerritory>
          </SiebelMessage>
        </Event>
```

```
          </ListOfEvent>

        <ns:LastEventId>20070316221509_Occam Territory_Occam Territory_1-
GF2YV_0_i.xml</ns:LastEventId>

        </ns:IntegrationEventWS_GetEvents_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

GetPicklistValues Method Samples

This topic contains samples for both regular and cascading picklists.

Sample of GetPicklistValues for a Cascading Picklist

This sample shows the output for a SubArea picklist that has a parent picklist of Area. The sample shows the related picklist values that correspond to parent picklist values of Installation (Server Crash and No Admin Login) and Maintenance (Need Upgrade and Need Patch).

NOTE: The last topic of the ParentPicklistValue element contains all the available values for the SubArea picklist.

```
<Input>

<Object>Account</Object>

<FieldName>SubArea</FieldName>

</Input>

<Output>

  <ListOfParentPicklistValue>

     <ParentPicklistValue>

        <Language>ENU</Language>

        <ParentFieldName>Area</ ParentFieldName>

        <ParentDisplayValue>Installation</ ParentDisplayValue>

        <ParentCode>Installation</ParentCode>

        <ListOfPicklistValue>

           <PicklistValue>

                  <DisplayValue>Server Crash</DisplayValue>

              <Code>Crash</Code>

           </PicklistValue>
           <PicklistValue>
                  <DisplayValue>No Admin Login</DisplayValue>
              <Code>NoLogin</Code>
           </PicklistValue>
        </ListOfPicklistValue>
     </ParentPicklistValue>
     <ParentPicklistValue>
     <Language>ENU</ Language>
     <ParentFieldName>Area</ ParentFieldName>
        <ParentDisplayValue>Maintenance</ ParentDisplayValue>
        <ParentCode> Maintenance </ ParentCode>
        <ListOfPicklistValue>
           <PicklistValue>
                 <DisplayValue>Need Upgrade</DisplayValue>
              <Code>Crash</Code>
           </PicklistValue>
           <PicklistValue>
                <DisplayValue>Need Patch</DisplayValue>
              <Code>NoLogin</Code>
           </PicklistValue>
        </ListOfPicklistValue>
     </ParentPicklistValue>
     <ParentPicklistValue>
     <Language>ENU</ Language>
     <ParentFieldName></ ParentFieldName>
        <ParentDisplayValue></ ParentDisplayValue>
        <ParentCode></ ParentCode>
        <ListOfPicklistValue>
           <PicklistValue>
                 <DisplayValue>Need Upgrade</DisplayValue>
```

```
             <Code>Crash</Code>

          </PicklistValue>

          <PicklistValue>

              <DisplayValue>Need Patch</DisplayValue>

             <Code>NoLogin</Code>

          </PicklistValue>

          <PicklistValue>

                 <DisplayValue>Server Crash</DisplayValue>

             <Code>Crash</Code>

          </PicklistValue>

          <PicklistValue>

                <DisplayValue>No Admin Login</DisplayValue>

             <Code>NoLogin</Code>

          </PicklistValue>

       </ListOfPicklistValue>

     </ParentPicklistValue>

     </ListOfParentPicklistValue>

</Output>
```

Sample of GetPicklistValues for a Regular Picklist

This sample shows the output for a Priority picklist that has possible values of High, Medium, and Low.

```
<Input>

<Object>Account</Object>

<FieldName>Priority</FieldName>

</Input>

<Output>

  <ListOfParentPicklistValue>

<ParentPicklistValue>

     <Language>ENU</ Language>

     <ParentFieldName></ ParentFieldName>
```

```
       <ParentDisplayValue></ ParentDisplayValue>

       <ParentCode></ ParentCode>

       <ListOfPicklistValue>

          <PicklistValue>

                <DisplayValue>High</DisplayValue>

             <Code>High</Code>

          </PicklistValue>

          <PicklistValue>

              <DisplayValue>Medium</DisplayValue>

             <Code>Medium</Code>

          </PicklistValue>

          <PicklistValue>

                 <DisplayValue>Low</DisplayValue>

             <Code>Low</Code>

          </PicklistValue>

       </ListOfPicklistValue>

    </ParentPicklistValue>

</ListOfParentPicklistValue>

</Output>
```

B 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, Activity, Campaign, Contact, Custom Object 01 - 03, Dealer, Lead, Opportunity, Service Request, and Solution.

NOTE: Attachments are not supported for the Web Services v2.0 API.

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 10MB in size just as they are for the Oracle CRM On Demand UI. Attachments greater than 10MB in size are therefore rejected. To accommodate Base64 encoding of binary files, which can result in a file up to 135% of the original file size, the request size limit for SOAP requests with attachment files is 14MB.

You can develop client applications that allows 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 358.
- **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">0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAAPgADAP7/
CQAGAAAAAAAAAAAAAAACAAAA+AAAAAAAAAAAEAAA+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>
```

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>
```

Index