

Oracle Web Services On Demand Guide

Version 8.0 (Oracle CRM On Demand Release 20)
June 2012

Copyright © 2005, 2012 Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Web Services On Demand

About Web Services 21

Web Services Core Technologies 21

Oracle CRM On Demand Web Services Toolkit 22

Oracle CRM On Demand Web Services and Integration with Oracle CRM On Demand 22

Web Services Security 23

Web Services Reliability 25

Web Services and the Oracle CRM On Demand Objects 25

Web Service APIs 25

About Parent-Child Relationships 29

Web Services On Demand and Custom Fields 29

Field Types Supported by Oracle CRM On Demand 30

Web Services v1.0 30

Web Services v2.0 31

Special Search Fields 32

Support for Multi-Select Picklists 32

Locale-Dependent Access to Oracle CRM On Demand 33

Validation of Email Fields 34

Mapping Primary Address Fields Using Web Services 35

Web Services Utilization 40

Web Service Client Name Identification 41

Supported Client Name Characters and Usage 41

Sending the Client Name in Stateless Web Services Requests 41

Sending the Client Name in Stateful Web Services 43

Web Services R16 Compatibility Mode 43

About Service Allotments 45

Determining Current Usage 45

Determining Historical Allotment Usage 46

When a Service Allotment Is Reached 46

Calculation of Allotment Usage 46

Chapter 3: Getting Started with Web Services

- Accessing Data in Oracle CRM On Demand 49
- Requesting Web Services Integration 49
 - Enabling Stateless Web Services Support 50
- Downloading WSDL files and Schema Files 50
- Incorporating WSDL Files into the Development Environment 52

Chapter 4: Establishing and Managing the Web Services Session

- About Establishing and Managing the Web Services Session 53
 - Oracle CRM On Demand Stateful and Stateless Request Support 53
- Using Stateless Web Service Requests 54
 - Enabling Support for Stateless Requests 54
 - Oracle CRM On Demand Stateless Authentication Mechanisms 54
 - Stateless Server Affinity Support 56
 - Maintenance of a Stateless Java Session 56
- Using Stateful Web Services Requests 57
 - Oracle CRM On Demand Stateful Authentication Mechanisms 58
 - Logout of the Web Services Session 61
- Integration Requests for the Web Services Session 61
- Comparison of Stateless Versus Stateful Web Services Transactions 62
- Single Sign-On 63
 - Outbound SSO 63
 - SSO Token Validation 63
 - Inbound SSO 64

Chapter 5: Best Practices for Designing Client Applications

- Best Practices for Integration Design 65
 - Sending of Web Services Requests to Oracle CRM On Demand 65
 - Flushing of Caches 65
 - Using Stateless Web Services 66
 - Setting and Querying Blank Values 68
 - Working with Opportunity Product Revenue Records 68
 - Error Handling and Logging 68
 - Handling Outages and Failures 69
 - Best Practices for Generating Web Services Proxy Classes in Java Environments 69
 - Avoiding Proxy Class Compilation Failure Due to the Java 64KB Limit 70
 - Entering Telephone Number Values through Web Services 74

Best Practices for Adhering to Web Service Allotments	76
Working with Record Ownership Modes	76
Best Practices for Integration Performance	77
Batch Processing	77
Session Management and Pooling	77
API Calls	78
Handling Bad Events in Integration Event Queues	81
Best Practices for Integration Management	81
Moving Customers Between Pods	81
Handling Outages	82
Maintaining SSL Certificates	82

Chapter 6: Web Services On Demand API Calls

Web Services On Demand API Calls	83
Web Services v1.0 API Calls	84
User Keys	84
The Echo Argument	85
Delete	85
DeleteChild	88
Insert	89
InsertChild	90
InsertOrUpdate	90
QueryPage	91
Update	101
UpdateChild	102
Web Services v2.0 API Calls	102
The Echo Argument	103
The ViewMode Argument	104
The LOVLanguageMode Argument	104
Delete	105
Execute	108
Insert	112
QueryPage	115
Update	136
Service API Calls	139
AuthAuthorizationUser	140
DeletedItemQueryPage	142
DeleteEvents	149
ExportWebServiceCreateExportRequest	151
ExportWebServiceMonitorExportRequest	154

ExportWebServiceRetrieveExportAttachment	157
GetEvents	159
GetLists	167
GetMapping	170
GetPicklistValues	173
GetServerTime	176
LoginHistoryQueryPage	178
MergeRecords	181
MetadataChangeSummaryQueryPage	184
SalesProcessQueryPage	187
SetPassword	193
SetSessionTimeZone	195
UpdatePicklist	197
UpdateCascadingPicklists	201
UserUsageQueryPage	206

Administrative Services API Calls 210

Access to the Administrative Services APIs for Nonadministrative Users	213
AccessProfileRead	214
AccessProfileReadAll	215
AccessProfileUpsert	215
ActionBarCustomWebAppletRead	216
ActionBarCustomWebAppletReadAll	217
ActionBarCustomWebAppletUpsert	218
ActionBarLayoutReadAll	219
ActionBarLayoutUpsert	220
CascadingPicklistRead	221
CascadingPicklistReadAll	222
CascadingPicklistUpsert	223
CurrencyRead	224
CurrencyReadAll	225
CurrencyUpsert	225
CurrentOrganizationRead	226
CustomRecordTypeRead	227
CustomRecordTypeReadAll	227
CustomRecordTypeUpsert	228
CustomWebLinkRead	229
CustomWebLinkReadAll	230
CustomWebLinkUpsert	231
CustomWebTabDelete	232
CustomWebTabRead	233
CustomWebTabReadAll	233
CustomWebTabUpsert	234

DetailPageCustomWebAppletRead	235
DetailPageCustomWebAppletReadAll	236
DetailPageCustomWebAppletUpsert	237
FieldManagementCreate	238
FieldManagementRead	239
FieldManagementReadAll	239
FieldManagementUpsert	240
HomepageLayoutReadAll	242
HomepageLayoutUpsert	243
HomepageCustomWebAppletRead	244
HomepageCustomWebAppletReadAll	245
HomepageCustomWebAppletUpsert	245
HomeTabCustomWebAppletRead	247
HomeTabCustomWebAppletReadAll	247
HomeTabCustomWebAppletUpsert	248
IndustryRead	249
IndustryReadAll	250
PageLayoutFieldRead	251
PageLayoutFieldReadAll	252
PageLayoutFieldUpsert	253
PageLayoutRelatedInformationRead	254
PageLayoutRelatedInformationReadAll	254
PageLayoutRelatedInformationUpsert	255
PageLayoutSectionReadAll	256
PageLayoutSectionUpsert	257
PicklistRead	258
PicklistReadAll	259
PicklistUpsert	260
RelatedInformationLayoutRead	261
RelatedInformationLayoutReadAll	262
RelatedInformationLayoutUpsert	263
RoleRead	264
RoleReadAll	265
RoleUpsert	265
SalesAssessmentTemplateRead	267
SalesAssessmentTemplateReadAll	268
SalesAssessmentTemplateUpsert	269
WorkflowActionSetRead	271
WorkflowActionSetReadForRuleSet	272
WorkflowActionSetUpsert	273
WorkflowRuleSetDelete	276
WorkflowRuleSetRead	277

WorkflowRuleSetReadAll 277

WorkflowRuleSetUpsert 278

Data Loader API Calls 280

BulkOpImportCreateRequest 280

BulkOpImportSendData 283

BulkOpImportGetRequestDetail 284

Chapter 7: Oracle CRM On Demand Objects Accessible Through Web Services

Reference Information About the Parent Objects 287

Oracle CRM On Demand User Keys 288

Audit Fields 288

Oracle CRM On Demand Status Keys 289

Oracle CRM On Demand Pick Maps 290

Filterable Fields 290

Core Parent Objects (Web Services v1.0) 290

Account 291

Activity 310

Asset 319

Book 322

Campaign 325

Contact 334

Current User 353

CustomObject1 - CustomObject3 355

Lead 364

Note 371

Opportunity 372

Product 385

Product Category 388

Service Request 390

Solution 394

Territory 398

User 399

User Group 402

Life Sciences Edition Parent Objects (Web Services v1.0) 405

MedEd 405

Financial Services Edition Parent Objects (Web Services v1.0) 408

Household 408

Portfolio 413

Automotive Edition Parent Objects (Web Services v1.0) 418

Dealer	418
Vehicle	421
Child Objects (Web Services v1.0)	424
Address	425
Attachment	426
Audit Trail	426
BookUser	426
Competitor	426
DelegatedUser	427
HouseholdTeam	427
Interests	428
Invitee	428
Login History	428
Multiple Contact Roles	428
OpportunityTeam	428
PortfolioTeam	429
ProductsDetailed	429
PromotionalItemsDropped	429
Quota	430
Recipient	430
Related Account	430
Related Contact	430
Revenue	430
SampleDropped	431
SubBook	431
Team	431
Core Parent Objects (Web Services v2.0)	431
Account	433
Activity	435
Address	440
Allotment Usage	442
Asset	443
Book	445
Campaign	447
Category	448
Contact	450
CustomObject	453
Exchange Rate	456
Group	458
Lead	459
Note	461
Opportunity	462

Product	465
Service Allotment	466
Service Request	468
Solution	470
Territory	472
User	474
Partner Relationship Management Edition Objects (Web Services v2.0)	475
Accreditation	476
Accreditation Request	478
Application	480
Certification	482
Certification Request	484
Course	486
Course Enrollment	488
Deal Registration	490
Exam	492
Exam Registration	494
Fund	496
MDF Request	498
Partner	500
Price List	502
Price List Line Item	503
Special Pricing Product	505
Special Pricing Request	506
Life Sciences Edition Parent Objects (Web Services v2.0)	509
Allocation	510
Business Plan	512
Blocked Product	514
Contact Best Time	515
Contact State License	517
Inventory Audit Report	519
Inventory Period	521
MedEd	522
Message Response	524
Messaging Plan	526
Messaging Plan Item	528
Messaging Plan Item Relation	530
Modification Tracking	532
Objective	533
Order	536
Order Item	537
Plan Account	539

Plan Contact	540
Plan Opportunity	542
Sample Disclaimer	544
Sample Inventory	545
Sample Lot	547
Sample Request	549
Sample Request Item	550
Sample Transaction	552
Signature	554
Transaction Item	555
Financial Services Edition Parent Objects (Web Services v2.0)	557
Claim	558
Coverage	560
Damage	562
Financial Account	563
Financial Account Holder	565
Financial Account Holding	566
Financial Plan	568
Financial Product	570
Financial Transaction	571
Household	573
Insurance Property	575
Involved Party	577
Policy	578
Policy Holder	580
Portfolio	582
Automotive Edition Parent Objects (Web Services v2.0)	584
Dealer	584
Vehicle	586
Child Objects (Web Services v2.0)	588
Account	590
Account Contact	591
Account Opportunity	593
Account Team	594
Activity	596
Address	598
Asset	600
Attachment	602
Attendee Call	602
Book	605
606	

Campaign	606
Competitor	608
Contact	609
Contact Role	610
Contact Team	612
Custom Object 1 through 3	613
Custom Object 4 through 15	614
Delegated User	615
Exchange Rate History	616
Interests	617
Invitee	618
Lead	619
Note	621
Opportunity	621
Opportunity Child	622
Opportunity Team	622
Products Detailed	624
Promotional Items Dropped	625
Recipient	626
Related Contact	627
Revenue	628
Sales Assessment	629
Sales Assessment Value	630
Sample Dropped	631
Service Request	632
Shared Address	634
User	637

Appendix A: Using Attachments with Web Services On Demand

About Attachments	639
The Attachment Element	639
Specifying File Attachment Content	641
Attaching Binary Data	641
Attaching a Text File	642
Attaching a File to Multiple Records	643
Retrieving or Querying File Attachments	645
Specifying URL Attachments	647

Appendix B: Mapping of Display Names to Service API Object Names

Mapping of Display Names 649

Appendix C: Web Services On Demand Limiters

Web Services On Demand Limiters 653

Web Services Request Rate 654

Index

1

What's New in This Release

What's New in Oracle Web Services On Demand Guide, Version 8.0 (Oracle CRM On Demand Release 20)

Table 1 lists changes described in this version of the documentation to support Version 8.0 (Oracle CRM On Demand Release 20) of the software.

Table 1. What's New in Oracle Web Services On Demand Guide, Version 8.0 (Oracle CRM On Demand Release 20)

Topic	Description
"Web Service Client Name Identification" on page 41	Modified topic. The <ClientName> parameter is now supported for the Administrative Services APIs, therefore the note stating that it is not supported has been removed.
"Oracle CRM On Demand Stateless Authentication Mechanisms" on page 54	Modified topic. For the Administrative Services APIs, the following login options are now supported for stateless Web services requests: <ul style="list-style-type: none"> ■ UserName and Password provided in SOAP security header (using WSSE Version 2.0 Namespace) ■ Oracle CRM On Demand Single Sign On (SSO) Token provided in SOAP security header
"Handling Bad Events in Integration Event Queues" on page 81	New topic. Information has been added about handling an integration event queue that contains bad events, that is, events that a client application cannot process.
"Web Services v2.0 API Calls" on page 102 "Service API Calls" on page 139	New topics. For the Web Services v2.0 and Service API methods, sample SOAP requests and SOAP responses are provided.
"GetEvents" on page 159 "Preparing the Integration Event WSDL File and Schema Files" on page 166	Modified topics. Information has been added about the merge and restore values for the operation attribute of ListOfEvent, which is returned by the GetEvents method.
"ExportWebServiceCreateExportRequest" on page 151 "ExportWebServiceMonitorExportRequest" on page 154 "ExportWebServiceRetrieveExportAttachment" on page 157	New topics. New methods for exporting data have been added to the Service API: <ul style="list-style-type: none"> ■ ExportWebServiceCreateExportRequest ■ ExportWebServiceMonitorExportRequest ■ ExportWebServiceRetrieveExportAttachment

Table 1. What's New in Oracle Web Services On Demand Guide, Version 8.0 (Oracle CRM On Demand Release 20)

Topic	Description
"ActionBarCustomWebAppletRead" on page 216 "ActionBarCustomWebAppletReadAll" on page 217 "ActionBarCustomWebAppletUpsert" on page 218	<p>New topics. A new Web service, Action Bar Custom Web Applet, has been added with the following methods:</p> <ul style="list-style-type: none"> ■ ActionBarCustomWebAppletRead ■ ActionBarCustomWebAppletReadAll ■ ActionBarCustomWebAppletUpsert
"CascadingPicklistUpsert" on page 223	<p>New topic. For the Cascading Picklists Web service, the CascadingPicklistUpsert method has been added.</p>
"DetailPageCustomWebAppletRead" on page 235 "DetailPageCustomWebAppletReadAll" on page 236 "DetailPageCustomWebAppletUpsert" on page 237	<p>New topics. A new Web service, Detail Page Custom Web Applet, has been added with the following methods:</p> <ul style="list-style-type: none"> ■ DetailPageCustomWebAppletRead ■ DetailPageCustomWebAppletReadAll ■ DetailPageCustomWebAppletUpsert
"HomepageLayoutUpsert" on page 243	<p>New topic. For the Homepage Layout Web service, the HomepageLayoutUpsert method has been added.</p>
"HomepageCustomWebAppletRead" on page 244 "HomepageCustomWebAppletReadAll" on page 245 "HomepageCustomWebAppletUpsert" on page 245	<p>New topics. A new Web service, Home Page Custom Web Applet, has been added with the following methods:</p> <ul style="list-style-type: none"> ■ HomepageCustomWebAppletRead ■ HomepageCustomWebAppletReadAll ■ HomeTabCustomWebAppletUpsert
"HomeTabCustomWebAppletRead" on page 247 "HomeTabCustomWebAppletReadAll" on page 247 "HomeTabCustomWebAppletUpsert" on page 248	<p>New topics. A new Web service, Home Tab Custom Web Applet has been added with the following methods:</p> <ul style="list-style-type: none"> ■ HomeTabCustomWebAppletRead ■ HomeTabCustomWebAppletReadAll ■ HomeTabCustomWebAppletUpsert
"PageLayoutSectionReadAll" on page 256 "PageLayoutSectionUpsert" on page 257	<p>New topics. A new Web service, Page Layout Section, has been added with the following methods:</p> <ul style="list-style-type: none"> ■ PageLayoutSectionReadAll ■ PageLayoutSectionUpsert

Table 1. What's New in Oracle Web Services On Demand Guide, Version 8.0 (Oracle CRM On Demand Release 20)

Topic	Description
"RelatedInformationLayoutRead" on page 261 "RelatedInformationLayoutReadAll" on page 262 "RelatedInformationLayoutUpsert" on page 263	<p>New topics. A new Web service, Related Information Layout, has been added with the following methods:</p> <ul style="list-style-type: none"> ■ RelatedInformationLayoutRead ■ RelatedInformationLayoutReadAll ■ RelatedInformationLayoutUpsert
"SalesAssessmentTemplateUpsert" on page 269	<p>New topic. For the Assessment Scripts Web service, the SalesAssessmentTemplateUpsert method has been added.</p>
"WorkflowActionSetRead" on page 271 "WorkflowActionSetReadForRuleSet" on page 272 "WorkflowActionSetUpsert" on page 273	<p>New topics. A new Web service, Workflow Action, has been added with the following methods:</p> <ul style="list-style-type: none"> ■ WorkflowActionSetRead ■ WorkflowActionSetReadForRuleSet ■ WorkflowActionSetUpsert
"WorkflowRuleSetDelete" on page 276 "WorkflowRuleSetRead" on page 277 "WorkflowRuleSetReadAll" on page 277 "WorkflowRuleSetUpsert" on page 278	<p>New topics. A new Web service, Workflow Rule, has been added with the following methods:</p> <ul style="list-style-type: none"> ■ WorkflowRuleSetDelete ■ WorkflowRuleSetRead ■ WorkflowRuleSetReadAll ■ WorkflowRuleSetUpsert
"Book" on page 322	<p>Modified topic. To support time-based book assignments, the StartDate, EndDate, and FuturePrimary fields are accessible through the Web Services v1.0 API for the Book child objects of Account and Contact.</p>
"Address" on page 440	<p>New topic. For Web Services v2.0 only, a new Web service, Address, has been added to allow querying of the shared address data for a company.</p> <p>This Web service is applicable only for companies that have the Enable Shared Addresses check box selected on the company profile.</p>
"Blocked Product" on page 514	<p>New topic. A new Web service, Blocked Product has been added to allow querying of the products that are blocked for particular contacts.</p>

Table 1. What's New in Oracle Web Services On Demand Guide, Version 8.0 (Oracle CRM On Demand Release 20)

Topic	Description
"Order" on page 536 "Order Item" on page 537	<p>New topics. New Web services, Order and Order Item, have been added to allow querying of orders and order items, when CRM On Demand Order Management is enabled for the company.</p>
"Sample Request" on page 549 "Sample Request Item" on page 550	<p>New topics. New Web services, Sample Request and Sample Request Item, have been added to allow querying of sample requests.</p>
"Attendee Call" on page 602 "Campaign" on page 606	<p>New topics. For Web Services v2.0 only, a number of new child objects are now accessible:</p> <ul style="list-style-type: none"> ■ Activity: Attendee call ■ Business plan: <ul style="list-style-type: none"> ■ MedEd ■ Messaging plan ■ Campaign: MedEd ■ Contact: <ul style="list-style-type: none"> ■ Blocked Product ■ Campaign ■ MedEd: Fund child
"Sales Assessment" on page 629 "Sales Assessment Value" on page 630	<p>New topics. For Web Services v2.0 only, the Sales Assessment and Sales Assessment Value child objects are now accessible for the following parent objects:</p> <ul style="list-style-type: none"> ■ Account ■ Business Plan ■ Objective
"Shared Address" on page 634	<p>New topics. For Web Services v2.0 only, the Shared Address child object is now accessible for the following parent objects:</p> <ul style="list-style-type: none"> ■ Account ■ Contact ■ Dealer ■ Partner
Appendix B, "Mapping of Display Names to Service API Object Names"	<p>Modified appendix. Details of some new objects have been added to the table showing the mapping of display names to the Service API object names.</p>

In addition, changes were made to improve the organization of the book and to eliminate the duplication of information.

2

Overview of Web Services On Demand

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

- [About Web Services](#)
- [Oracle CRM On Demand Web Services and Integration with Oracle CRM On Demand on page 22](#)
- [Web Services and the Oracle CRM On Demand Objects on page 25](#)
- [Field Types Supported by Oracle CRM On Demand on page 30](#)
- [Web Services Utilization on page 40](#)
- [About Service Allotments on page 45](#)

About Web Services

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

Web Services Core Technologies

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

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

For more information on Web services technologies, see:

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

Oracle CRM On Demand Web Services Toolkit

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

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

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

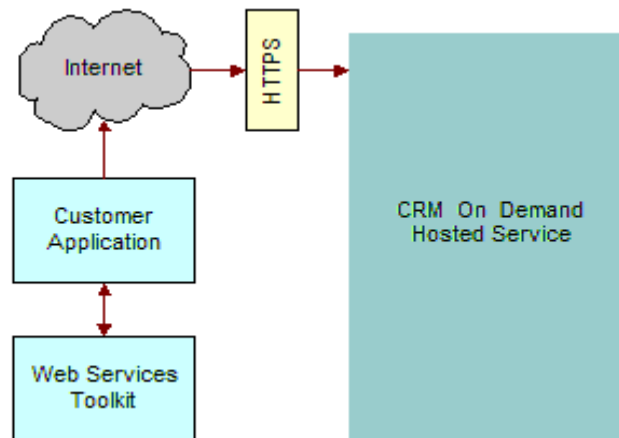


Figure 1. How Web Services Communicate with Oracle CRM On Demand

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

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

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

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

Web Services Security

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

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

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

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

Support for the WS-I Basic Security Profile Version 1.0

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

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

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

WSSE Namespace Support

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

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

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

For more information about stateless transactions and the use of the WSSE namespace, see [Chapter 4, “Establishing and Managing the Web Services Session.”](#)

Web Services Reliability

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

Web Services and the Oracle CRM On Demand Objects

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

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

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

Web Service APIs

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

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

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

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

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

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

Table 2 shows the methods available through the Web Services v1.0 and Web Services v2.0 APIs for access to objects.

Table 2. Web Services v1.0 and Web Services v2.0 Methods

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

Table 3 shows differences between Web Services v1.0 and Web Services v2.0.

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

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

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

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

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

Web Services v1.0	Web Services v2.0
DeleteChild call is used to delete the child records	<p>For Web Services v2.0:</p> <ul style="list-style-type: none"> ■ Delete call can be used to delete both parent records and child records ■ If the child node specified in the request is available, the Delete call deletes the child in the existing parent record, and leaves the parent record undeleted. ■ If the child node is missing, the Delete call deletes the existing parent record.
InsertChild, UpdateChild, and DeleteChild methods are used to perform operations on child records	<p>In an Execute request, a specific node within the request can be skipped using the "operation=skipnode" attribute.</p> <p>This can be used to simulate InsertChild, UpdateChild or DeleteChild by skipping the parent node and only performing the specified actions on the child records.</p>
LOVLanguageMode argument is not available	The LOVLanguageMode argument is an input argument for all of the Web Services v2.0 calls. It determines whether the processing for picklist fields occurs using language independent codes (LIC) or language dependent codes (LDC).
ViewMode argument is not available	The ViewMode argument, which specifies the level of access to records specified in the method call, is available for all of the Web Services v2.0 calls

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

- In the Web Services v2.0 API, strong data typing is supported. Therefore, in the Web Services v2.0 WSDL files, fields are represented by a range of xsd: data types, while in Web Services v1.0 WSDL files, all fields have the xsd:string data type. For more information, see ["Field Types Supported by Oracle CRM On Demand" on page 30](#).
- In Web Services v2.0, messages do not include the business service name, and have the format: `[ObjectName][Method]_[Input/Output]`
 For example:
 AccountInsert_Input, ContactQueryPage_Output
 as opposed to the following for Web Services v1.0:
 AccountWS_AccountInsert_Input, ContactWS_ContactQueryPage_Output
- The target namespace of the WSDL for Web Services v2.0 is:

urn: crmondemand/ws/ecbs/objectname/

compared to the following for Web Services v1.0:

urn: crmondemand/ws/objectname/

About Parent-Child Relationships

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

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

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

Web Services On Demand and Custom Fields

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

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

Custom Fields can be referenced using two different integration tags:

- 1 The Custom WSDL file uses the format:

fieldtypeDisplay_Name

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

- 2 The Generic WSDL file uses the format:

fieldtype##

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

To view or modify integration tag information for a record type

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

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

- 2 Click Advanced.

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

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

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

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

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

For more information about downloading WSDL files, see *Oracle CRM On Demand Online Help*.

Field Types Supported by Oracle CRM On Demand

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

Web Services v1.0

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

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

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

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

Web Services v2.0

The Web Services v2.0 API supports strong data types for fields, so fields are represented by appropriate XSD data types. [Table 4](#) shows the list of supported XSD data types.

Table 4. Data Type Mapping in the Web Services v2.0 API

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

If an incorrect data type is provided in a Web services request, the field is updated to NULL or a default value for that specific data type, as shown in [Table 5](#).

Table 5. Updating of Fields When Incorrect Data Types are Provided in the Web Services v2.0 API

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

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

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

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

Special Search Fields

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

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

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

Support for Multi-Select Picklists

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

- Account
- Activity
- Contact
- Custom Object 01

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

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

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

Locale-Dependent Access to Oracle CRM On Demand

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

Date and Time Fields

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

MM/DD/YYYY hh: mm: ss

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

The XSD dateTime datatype has the format:

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

For example:

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

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

As a further example:

2009-10-09T17:00:00Z

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

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

Number and Currency Fields

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

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

Validation of Email Fields

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

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

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

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

Mapping Primary Address Fields Using Web Services

In Web services requests, a *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. The primary address field in Oracle CRM On Demand can vary depending on the Country value for each address; thus the *PrimaryAddressLine1* field will map to a different field in the address object based on the Country value. [Table 6](#) and [Table 7](#) shows the mapping for the *PrimaryAddressLine1* field depending on the selected Country value.

NOTE: The actual name of the *PrimaryAddressLine1* fields varies with the record type as shown in “[Objects Supporting the PrimaryAddressLine1 Field](#)” on page 39.

Table 6. *PrimaryAddressLine1* To Address Field Mapping by Country

Address	Lead	Account (Billing Address)	Account (Shipping Address)	Contact (Account Address)	Contact (Contact Address)	Country
Street Address	Street Address	Bill To Street Address	Ship To Street Address	Personal Street Address	Primary Street Address	Group A See Table 7
Street Address 3	Street Address 3	Bill To Street Address 3	Ship To Street Address 3	Personal Street Address 3	Primary Street Address 3	Nauru
Postal Code	Postal Code	Bill To Postal Code	Ship To Postal Code	Personal Postal Code	Primary Postal Code	Group B See Table 7

Table 6. *PrimaryAddressLine1* To Address Field Mapping by Country

Address	Lead	Account (Billing Address)	Account (Shipping Address)	Contact (Account Address)	Contact (Contact Address)	Country
County	County	Bill To County	Ship To County	Personal County	Primary County	Group C See Table 7
Province	Province	Bill To Province	Ship To Province	Personal Province	Primary Province	Qatar
City	City	Bill To City	Ship To City	Personal City	Primary City	Papua New Guinea

Table 7. Groups of Countries with Different Address Field Mappings

Group	Countries
A	United States and all other countries apart from those in groups B and C, and those mentioned in Table 6 .
B	Hungary, Belarus, Burkina Faso, Congo, Kazakhstan, Kyrgyzstan, Russian Federation, Congo Sudan, Turkmenistan, Ukraine
C	Antigua and Barbuda, Benin, Burundi, Botswana, Cameroon, Central African Republic, Chad, Comoros, Djibouti, Equatorial Guinea, Ethiopia, Gabon, Ghana, Guinea, Ivory Coast, Kenya, Lesotho, Malawi, Mauritania, Namibia, Niger, Niue, Oman, Puerto Rico, Rwanda, Seychelles, Solomon Islands, Swaziland, Tanzania, Togo, Tonga, Tuvalu, Uganda, United Arab Emirates, Vanuatu

Querying for an Address Record using *PrimaryAddressLine1*

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

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

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

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

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

```
<Account>
```

```
<Location>Toronto</Location>
```

```

<AccountName>ACCOUNTTEST1</AccountName>
<ListOfAddress Lastpage="true">
  <Address>
    <Id>1QA2-R7C30</Id>
    <StreetAddress3></StreetAddress3>
    <Country>Canada</Country>
    <County></County>
    <Description></Description>
    <Province>ON</Province>
    <ZipCode>M2H 3G5</ZipCode>
    <City>Toronto</City>
    <IntegrationId>1QA2-R7C30</IntegrationId>
    <Address>100 Main Street</Address>
    <StreetAddress2></StreetAddress2>
    <PrimaryAddressLine1>100 Main Street</PrimaryAddressLine1>
  </Address>
  ...
</ListOfAddress>
</Account>
</ListOfAccount>
</ns: AccountQueryPage_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

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

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV: Envelope xmlns: SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns: xsd="http://www.w3.org/
  2001/XMLSchema">
  <SOAP-ENV: Body><ns: AccountQueryPage_Output xmlns: ns="urn: crmondemand/ws/ecbs/account/"
  ">
    <ListOfAccount xmlns="urn: /crmondemand/xml /Account/Data" Lastpage="true">

```

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

Inserting or Updating an Address Record using *PrimaryAddressLine1*

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

This is shown in [Table 8](#).

Table 8. Value specified for *PrimaryAddressLine1* and Primary Address Field

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

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

Table 9. Value specified for *PrimaryAddressLine1* only

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

Objects Supporting the *PrimaryAddressLine1* Field

The *PrimaryAddressLine1* field is available on a number of objects accessible through the Web Services v2.0 interface as shown in [Table 10](#) and [Table 11](#).

Table 10. Parent Objects on Which the *PrimaryAddressLine1* field is available

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

Table 11. Child Objects on Which the *PrimaryAddressLine1* field is available

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

Web Services Utilization

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

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

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

For more information about the Web Service Client Name parameter, see [“Web Service Client Name Identification” on page 41](#).

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

You can also use the `UserUsageQueryPage` method to retrieve information about Web services utilization. For more information about this method, see [“UserUsageQueryPage” on page 206](#).

See *Oracle CRM On Demand Online Help* for more information on using the Web Services Utilization page.

Web Service Client Name Identification

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

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

Supported Client Name Characters and Usage

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

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

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

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

`[Developer], [Client Name]`

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

`XYZ Consulting, Account Synchronization Utility`

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

Sending the Client Name in Stateless Web Services Requests

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

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

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

  </soap:Header>

  <soap:Body>

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

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

        <Account>

          <AccountName>LIKE 'a1' </AccountName>

          <Location/>

        </Account>

      </ListOfAccount>

    </AccountQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

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

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

Sending the Client Name in Stateful Web Services

A stateful Web service request execution involves:

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

For a stateful request:

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

Stateful Login

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

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

For a login with username and password:

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

Http Header:

username: <username>

password: <password>

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

For an SSO login:

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

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

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

Web Services R16 Compatibility Mode

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

- The client name specified in the SOAP Header is used for the login operation and stateful operation execution

- With the returned session ID, for subsequent requests, if the client name is specified in the SOAP header, it is ignored.
- As for stateful requests, the client name with which login occurs (that is, the first SOAP request in this case) is displayed in the Web Services Utilization page with all requests for the stateful cycle.

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

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

```
<soap:Header>
```

```
  <wsse:Security>
```

```
    <wsse:UsernameToken>
```

```
      wsse:Username>USERNAME</wsse:Username>
```

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

```
    </wsse:UsernameToken>
```

```
  </wsse:Security>
```

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

```
</soap:Header>
```

```
<soap:Body>
```

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

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

```
      <Account>
```

```
        <AccountName>LIKE 'a1' </AccountName>
```

```
        <Location/>
```

```
      </Account>
```

```
    </ListOfAccount>
```

```
  </AccountQueryPage_Input>
```

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

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

About Service Allotments

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

The service allotments for Web service usage include the following:

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

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

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

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

NOTE: See *Oracle CRM On Demand Online Help* for more information about service allotment administration.

Determining Current Usage

The Web Services Operations allotment is measured using a 24-hour rolling window. Current usage is displayed in the Oracle CRM On Demand UI or can be retrieved using the Service Allotment Web service (see [“Service Allotment” on page 466](#)). Current usage reflects the usage for the current hour plus the previous 23 hours.

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

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

Determining Historical Allotment Usage

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

- The Allotment Usage Web service (see [“Allotment Usage” on page 442](#))
- The Export Assistant
- The List Management Export feature in the Service Allotment Usage History page under Company Administration.

When a Service Allotment Is Reached

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

See *Oracle CRM On Demand Online Help* for more information on configuring email alerts for service allotments.

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

For information about best practices, see [“Best Practices for Adhering to Web Service Allotments” on page 76](#).

Calculation of Allotment Usage

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

Web Services Operations Allotment

The Web service operation count is incremented whenever a Web service request is received and executed. A single Web service SOAP request, when processed, might result in one or more Web service operations being executed. For example, [Table 12](#) shows the number of operations resulting for different types of request.

Table 12. Examples of Number of Operations for Different Web Services Requests

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

Table 12. Examples of Number of Operations for Different Web Services Requests

Type of Request	Number of Operations
Contact update request containing 10 Contact records (with no child operations)	10
Account update request containing a single Account record with 3 Account Team records	4
Query operations	
Simple query for a set of Accounts	1
Query for a set of Accounts and the associated Contacts for each Account	$1 + n$, where: <ul style="list-style-type: none"> ■ 1 operation to retrieve the set of n Accounts matching the specified filter criteria ■ n operations to retrieve the set of Contacts associated with each Account (1 operation for each Account)

Web Services Concurrent Request Allotment

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

3

Getting Started with Web Services

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

- [Accessing Data in Oracle CRM On Demand on page 49](#)
- [Requesting Web Services Integration on page 49](#)
- [Downloading WSDL files and Schema Files on page 50](#)
- [Incorporating WSDL Files into the Development Environment on page 52](#)

Accessing Data in Oracle CRM On Demand

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

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

Requesting Web Services Integration

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

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

- **Enable Web Services Access.**

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

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

- **Manage Access to Web Services.**

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

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

- 1 Contact Customer Care and request that the Administrator (or equivalent custom role) be granted the Manage Access to Web Services Integration privilege.
- 2 Configure roles. The Administrator grants or revoke access to Web services on for each role.

Enabling Stateless Web Services Support

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

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

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

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

For more information about these namespaces, see [“WSSE Namespace Support” on page 24](#).

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

Downloading WSDL files and Schema Files

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

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

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

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

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

To download a WSDL file

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

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

- 4 Click either the Download Custom WSDL or Download Generic WSDL button to open a popup window containing the selected WSDL file.
- 5 Save the WSDL file to your computer.

Downloading Schema Files

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

If you create custom fields or rename fields for a record type, you cannot use the Generic XSD files for tracking these fields. Instead, you must download a Custom XSD file using the Download Custom Schema button in the Web Services Administration page. For more information about the Integration Events Web service, see [“Preparing the Integration Event WSDL File and Schema Files” on page 166](#).

To download a Schema file

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

- 3 From the WSDL Object drop-down list, select the required record type, for example, Account.
The objects displayed in the WSDL Object drop-down list depend on the record types that are set up for your company.
- 4 Click one of the following buttons:
 - Download Custom Schema
 - Download Generic SchemaA page containing the Schema is displayed. Depending on whether you select Download Custom Schema, or Download Generic Schema, custom fields are displayed differently in the Schema. For Custom Schema, the XML tags for the custom fields are based on the Integration tags from Field Setup. For Generic Schema, the custom fields are based on generic XML tags.

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

Incorporating WSDL Files into the Development Environment

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

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

4

Establishing and Managing the Web Services Session

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

- [About Establishing and Managing the Web Services Session on page 53](#)
- [Using Stateless Web Service Requests on page 54](#)
- [Using Stateful Web Services Requests on page 57](#)
- [Integration Requests for the Web Services Session on page 61](#)
- [Comparison of Stateless Versus Stateful Web Services Transactions on page 62](#)
- [Single Sign-On on page 63](#)

About Establishing and Managing the Web Services Session

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

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

Oracle CRM On Demand Stateful and Stateless Request Support

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

- Stateless requests
- Stateful session management

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

Stateful Web services transactions require the client application to maintain and provide a session identifier (a JSESSIONID value) to perform multiple requests to Oracle CRM On Demand using the same session. Subsequent requests with the session identifier do not require reauthentication.

Using Stateless Web Service Requests

Interactive applications can be developed such that a user's credentials are supplied within a request sent to Oracle CRM On Demand, eliminating the need for an explicit login request. If the request qualifies as a stateless request (see [“Enabling Support for Stateless Requests” on page 54](#)), the Oracle CRM On Demand server checks to see whether a session has already been established for that user. If a session is found, it is reused for the new request. If no existing session is found for the user, a new session is established.

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

Enabling Support for Stateless Requests

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

- A company must have stateless Web services support enabled, see [“Enabling Stateless Web Services Support” on page 50](#).
- The WS-I login mechanism must be used with the correct namespace and formats as described by the WS-I basic security profile at <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html#UsernameToken>:
 - The WSSE security namespace (<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd>) must be specified in the request.

Requests that meet these criteria qualify as stateless requests.

Oracle CRM On Demand Stateless Authentication Mechanisms

Stateless login is available on all APIs. Stateless Web services requests for the Web Services v1.0, Web Services v2.0, Services, and Data Loader APIs can be authenticated using:

- UserName and Password provided in SOAP security header (using WSSE Version 1.0 Namespace)
- Oracle CRM On Demand Single Sign On (SSO) Token provided in SOAP security header
- SSO with SAML v1.1 or v2.0

For the Administrative Services APIs only, the following login options are supported for stateless Web services requests:

- UserName and Password provided in SOAP security header (using WSSE Version 2.0 Namespace)

- Oracle CRM On Demand Single Sign On (SSO) Token provided in SOAP security header

Login with UserName and Password in the SOAP Security Header

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

```
<soap: Header>

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

    <wsse: UsernameToken>

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

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

    </wsse: UsernameToken>

  </wsse: Security>

</soap: Header>
```

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

For the Administrative Services API, a similar login mechanism is used, but with a WSSE Version 2.0 namespace instead of the WSSE Version 1.0 Namespace.

Login with Oracle CRM On Demand Single Sign-On Token in the SOAP Header

This login mechanism is a type of outbound SSO, see [“Outbound SSO” on page 63](#).

The client application supplies the SSO token in the <wsse:KeyIdentifier> element of the SOAP header:

```
<soap: Header>

  <wsse: Security>

    <wsse: SecurityTokenReference>

      <wsse: KeyIdentifier Value="http://schemas.crmondemand.com/ws/2011/01/secext#SS0TokenKeyIdentifier">$6$qx6pJ/czNw01trwQRazQ26j 4osNi QHMoqQSwRfpz/6HX2D5cw=; $6$Ij wK0/BB0BW5oi uqC7P/Txw0BX1LxVpExR9vp7P5J/ki xzGFWI j xHyRye7zy9Ld2g2vKp4W4j ykxj bgF3KE8CF0GmD5g==</wsse: KeyIdentifier>

    </wsse: SecurityTokenReference>

  </wsse: Security>
```

</soap: Header>

NOTE: URL encoding of the SSO token is not supported when provided in the SOAP security header. If the SSO token is URL encoded, it must first be decoded before supplying it in the SOAP security header.

The SOAP request will not be processed if the SSO token expired. It is best practice to validate the SSO token before using it for login, see [“SSO Token Validation” on page 63](#).

SSO with SAML v1.1 or v2.0

This login mechanism is a type of inbound SSO, see [“Inbound SSO” on page 64](#).

For SSO using Security Assertion Markup Language (SAML), Oracle CRM On Demand only supports the SAML Web Browser Profiles - the Browser/Artifact Profile and the Browser/POST Profile using the Proprietary Token method.

Stateless Server Affinity Support

The HTTP response of a Stateless Web Service request includes a cookie, containing a session ID (JSESSIONID). This session ID does not identify a specific user session, but rather is used to identify the server in the Oracle CRM On Demand environment on which the session was initially created. Subsequent requests can be sent to include the session ID value, thus ensuring that the request is routed to the same server as the original request, maintaining server affinity. The actual session, however, cannot be used to process the request unless the session is explicitly kept active, see [“Maintenance of a Stateless Java Session” on page 56](#).

NOTE: Providing a session ID in subsequent requests is optional, however it is recommended that this value is included as it will increase performance when there are multiple application servers deployed.

Maintenance of a Stateless Java Session

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

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

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

The following is an example of a SOAP header containing the <SessionKeepAlive> element:

```
<soap: Header>

  <crmod: Sessi onKeepAl i ve>true</crmod: Sessi onKeepAl i ve>

  <wsse: Securi ty>

    <wsse: Securi tyTokenReference>

      <wsse: KeyI denti fi er

        Val ueType="http://schemas.crmondemand.com/ws/2011/01/
        secext#SS0TokenKeyI denti fi er">$6$qx6pJ/rGFI DyJBy91C3Uxi fc3oDEg/
        Sshtrsyj 2z27RoHm05M=; $6$j Wkqu/
        Hvl Xoqw2FoHx10GDcYDc5WdZ2UV4v1G06AMEBNbWZv0JZi nyJ1C4v70HJR400VhZ4J3X+U/
        EvrP1c82LYHV0HA==</wsse: KeyI denti fi er>

      </wsse: Securi tyTokenReference>

    </wsse: Securi ty>

  </soap: Header>
```

Using Stateful Web Services Requests

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

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

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

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

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

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

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

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

Once a session ID is retrieved, it is possible to use the session ID to invoke Web services by using HTTP POST; for more information, see ["Integration Requests for the Web Services Session" on page 61](#).

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

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

Oracle CRM On Demand Stateful Authentication Mechanisms

Stateful login can be used with the Web Services v1.0, Web Services v2.0, Service, and Data Loader APIs, but is not available for the Administrative Services APIs.

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

- HTTP Login (containing UserName and Password in the HTTP Header)
- UserName and Password provided in SOAP security header (using WSSE Draft Namespace)
- HTTP Login (with Oracle CRM On Demand tSingle Sign On (SSO) Token in HTTP Header)
- SSO using SAML v1.1

Login Using HTTP GET (Providing UserName and Password in the HTTP Header)

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

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

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

NOTE: The login parameter value is case sensitive.

Login Input

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

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

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

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

Login Output

The login command returns the following items:

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

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

Login with UserName and Password in the SOAP Security Header

A similar login mechanism to that used for stateless requests is used, but with a WSSE draft namespace instead of the WSSE Version 1.0 Namespace. For more information, see ["Login with UserName and Password in the SOAP Security Header" on page 55](#).

The SOAP header contains the element `<wsse:UsernameToken>`, which has child elements containing a username and password:

```
<soap: Header>

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

    <wsse: UsernameToken>

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

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

    </wsse: UsernameToken>

  </wsse: Security>

</soap: Header>
```

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

Login with Oracle CRM On Demand Single Sign-On (SSO) Token in the HTTP Header

This login mechanism is a type of outbound SSO, see [“Outbound SSO” on page 63](#).

The client instantiates an Oracle CRM On Demand Web services session and obtains a valid session ID by sending an HTTP GET request to the following URL and specifying the SSO token:

```
https://secure-ausomx[ENV].crmondemand.com/Services/
Integration?command=ssologin&odSsoToken=[SSOTOKEN]
```

where:

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

NOTE: The login parameter value is case sensitive.

- [SSOTOKEN] is the SSO token string used to authenticate a user to have access to data within Oracle CRM On Demand.

If the login request is successful, the server returns the session ID in the response.

The session will not be instantiated and the session ID will not be returned if the SSO token has expired. It is best practice to validate the SSO token before using it for login, see [“SSO Token Validation” on page 63](#).

SSO with SAML v1.1

For SSO using Security Assertion Markup Language (SAML), Oracle CRM On Demand only supports the SAML Web Browser Profiles – the Browser/Artifact Profile and the Browser/POST Profile. For information about logging in and retrieving the session ID to be used for stateful requests, see [“Inbound SSO” on page 64](#)

Logout of the Web Services Session

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

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

NOTE: The parameter value `logoff` is case sensitive.

Logoff Input

The preferred method is to include the JSESSIONID with the request as a cookie with the same name. There are no other URL parameters or HTTP headers, and there is no HTTP body.

Logoff Output

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

Alternative Logoff

The JSESSIONID can also be included in the URL instead of a cookie. The following URL can be used, where XXXX is the JSESSIONID.

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

Integration Requests for the Web Services Session

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

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

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

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

```
https://secure-ausomx[ENV].crmondemand.com/Services/Integration/object;  
j sessionid=xyZ12489w3482413
```

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

Integration request output. The properties returned by the HTTP server populate the response headers and the response body. Table 13 shows the top-level properties that specify key properties of the HTTP response.

Table 13. Properties of the HTTP Response

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

Comparison of Stateless Versus Stateful Web Services Transactions

Stateful Web services are generally used when querying a large number of records or performing bulk data operations, for example uploading large volumes of data. Some characteristics of stateful Web services transactions include:

- Do not utilize OM session pooling (the client application must manage the session for each request)
- Return a valid JSESSIONID value, which is used for authenticating subsequent requests
- An OM session is not shared and can store details about previous requests, this is helpful when performing pagination over a query result set. The session will not be used by other applications as the session ID must be explicitly specified in the requests.
- The session limit quota controls the number of sessions that can be established for a company.

In general, it is recommended that customers use stateless Web services for several important reasons:

- Stateless Web services provide more efficient resource utilization for most uses.
- Stateless Web services do not impact the session limit quota.
- Since customers are responsible for Web service development, stateless Web services can dramatically reduce complexity and cost of developing new integrations as resource management is performed by Oracle. For stateful Web services, this is the customer's responsibility. For example, customers do not need to code logic to log off following a stateless request/response.
- Stateless Web service requests achieve a higher level of concurrency because Oracle CRM On Demand code manages sessions.

- Additional error handling logic is also the customer's responsibility and must be implemented correctly with stateful Web services.

Customers must ensure stateless Web services are used for short, interactive requests where high concurrency is required.

Some characteristics of stateless Web services transactions include:

- Support OM session pooling. Sessions are managed and pooled internally by the Oracle CRM On Demand server.
- Perform a logoff (invalidation of the Java session) after processing each request unless the <SessionKeepAlive> parameter is set to true. The client application does not have to manually submit a request to log off a session.
- Return a JSESSIONID value to help maintain server affinity (that is, direct subsequent requests to the same Java server), if the <SessionKeepAlive> parameter is specified in the request.

Stateless Web services transactions must be used for Administrative Services APIs. A full login is performed for each request.

Single Sign-On

The Single Sign-On (SSO) feature of Oracle 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.

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.

For security reasons, the SSO token is only valid for a short period of time.

SSO Token Validation

The following steps are used to validate an SSO token:

- 1 The third-party application can pass the SSO token as an URL parameter called "odSsoToken" of the following URL through a POST or GET request:

`https://server/Services/SSOTokenValidate?odSsoToken = [SSOTOKEN]`

2 If successful, the server returns the username in the HTTP body.

The server returns errors if the SSO token has expired or is missing from the request.

Inbound SSO

Inbound SSO allows a company that has its own internal authentication system to validate a user and then seamlessly allow the user to log in to Oracle CRM On Demand.

The company must be set up to use SSO for Oracle CRM On Demand:

- The company and optionally the user must be configured for SSO.
- External Identifier for SSO must be specified.
- ITS URL for SSO must be specified.

If your company has been set up to use SSO for Oracle CRM On Demand, the following steps are used to initiate a Web service request to Oracle CRM On Demand using SSO:

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

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

2 The server returns the SSO ITS URL in the "X-SsoItsUrl" HTTP header of the response.

3 The Web service makes a request with the ITS URL.

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

NOTE: The behavior of the identity provider is beyond the control of Oracle CRM On Demand.

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

- Follow redirects
- Accept cookies

Customers are responsible for ensuring that their client applications are compatible with their chosen identity provider. For code samples for single sign-on see: https://codesamples.samplecode.oracle.com/servlets/Scarab/action/ExecuteQuery?query=crm_on_demand

5

Best Practices for Designing Client Applications

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

- [Best Practices for Integration Design on page 65](#)
- [Best Practices for Integration Performance on page 77](#)
- [Best Practices for Integration Management on page 81](#)

Best Practices for Integration Design

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

Sending of Web Services Requests to Oracle CRM On Demand

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

Flushing of Caches

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

For stateless Web services only, users can flush the caches to reload any metadata information that has changed recently. You use the `MetadataChangeSummaryQueryPage` method to determine whether there have been any metadata changes; see [“MetadataChangeSummaryQueryPage” on page 184](#).

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

Best Practices for Flushing of Caches

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

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

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

Usage

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

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

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

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

  </soap:Header>

  <soap:Body>

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

  </soap:Body>

</soap:Envelope>
```

Using Stateless Web Services

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

NOTE: The best practices for stateful Web services operations must also be applied to the use of stateless Web services; see [“Session Management and Pooling”](#) on page 77.

Using Both Stateful and Stateless Web Services

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

For example, if you follow this sequence:

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

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

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

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

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

Avoid Multiple Concurrent Requests

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

- If possible, try the request again later, as the load on the server might decrease.
Most load conditions are rare and temporary. You might never see one during development.
- Retry with an exponential backoff. The client can be implemented such that retries are issued automatically using an exponential retry rate (that is, retries occur after 100ms, 200ms, 400ms, 1600ms and so on).

Use Sort Criteria when Using QueryPage

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

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

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

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

Setting and Querying Blank Values

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

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

Working with Opportunity Product Revenue Records

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

Error Handling and Logging

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

- Logging of detailed information about the error observed.
- Logging of the body and header information of all SOAP requests and responses. For the resolution of some errors, the actual SOAP request can be extremely useful in identifying the root cause of a problem.
- A call stack, which can be extremely important when analyzing problems and can provide useful hints that might 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

To handle the errors, a client application needs to be able to understand the SOAP fault that is returned in the SOAP response when an error is encountered in processing a Web service request.

For information about SOAP faults and error codes returned by Oracle CRM On Demand, see *Oracle Web Services On Demand Troubleshooting Guide*.

Handling Outages and Failures

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

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

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

Best Practices for Generating Web Services Proxy Classes in Java Environments

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

Oracle JDeveloper Workaround

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

To work around the failure

- 1** Avoid proxy class compilation failure.
See [“Avoiding Proxy Class Compilation Failure Due to the Java 64KB Limit”](#) on page 70.
- 2** Increase the Java heap size in the `jdev.conf` file.

Axis Workaround

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

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

To work around the failure

- 1** Remove unused child objects from WSDL files.
- 2** Increase the Java heap size in the `wsdl2java.bat/wsdl2java.sh` file.

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

- 1** In the `<Types>` section in the WSDL, find the `<xsd:schema>` element with the target namespace `"urn:/crmondemand/xml/ParentName/Query"`.
 - a** Traverse to the `<xsd:complexType>` element with the name `"ParentNameQuery"`.
 - b** Remove the `<xsd:element>` element with the name `"ListOfChildNameQuery"`.
 - c** Remove the `<xsd:complexType>` element with the name `"ListOfChildNameQuery"`.
 - d** Remove the `<xsd:complexType>` element with the name `"ChildNameQuery"`.
- 2** Find the `<xsd:schema>` element with the target namespace `"urn:/crmondemand/xml/ParentName/Data"`.
 - a** Traverse to the `<xsd:complexType>` element with the name `"ParentNameData"`.
 - b** Remove the `<xsd:element>` element with the name `"ListOfChildNameData"`.
 - c** Remove the `<xsd:complexType>` element with the name `"ListOfChildNameData"`.
 - d** Remove the `<xsd:complexType>` element with the name `"ChildNameData"`.

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

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

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

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

- 2 Compile the proxy classes.

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

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

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

There are two possible workarounds:

- Split the doSerialize method.

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

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

- Delete unused fields generated in the serialize methods.

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

Figure 2. A doSerialize() Method Before Splitting:

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

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

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

    if (instance.getModifiedDate() != null) {
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );

myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getModifiedDate(),
    ns2_ModifiedDate_QNAME, null, writer, context);
    }

    if (instance.getCreatedDate() != null) {
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );

myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getCreatedDate(),
    ns2_CreatedDate_QNAME, null, writer, context);
    }
}
```

```

        if (instance.getModifiedByld() != null) {
            myns3_string__java_lang_String_String_Serializer.setNullable( false );
            myns3_string__java_lang_String_String_Serializer.serialize(instance.getModifiedByld(),
                ns2_ModifiedByld_QNAME, null, writer, context);
        }
        if (instance.getCreatedByld() != null) {
            myns3_string__java_lang_String_String_Serializer.setNullable( false );
            myns3_string__java_lang_String_String_Serializer.serialize(instance.getCreatedByld(),
                ns2_CreatedByld_QNAME, null, writer, context);
        }
        if (instance.getModld() != null) {
            myns3_int__java_lang_Integer_Int_Serializer.setNullable( false );
            myns3_int__java_lang_Integer_Int_Serializer.serialize(instance.getModld(),
                ns2_Modld_QNAME, null, writer, context);
        }
        ...
    }

```

Figure 3. A doSerialize() Method After Splitting

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

```

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

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

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

```

public activity.proxy.types.crmondemand.xml.activity.data.ActivityData
doSerialize1(activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance, XMLWriter writer,
SOAPSerializationContext context) throws Exception

```

    {
        if (instance.getModifiedDate() != null)
        {
            myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );
            myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getModifiedDate(),
                ns2_ModifiedDate_QNAME, null, writer, context);
        }
    }

```



```

    }
    if (instance.getCreatedDate() != null)
    {
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.setNullable( false );
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializer.serialize(instance.getCreatedDate(),
            ns2_CreatedDate_QNAME, null, writer, context);
    }
    ...
return instance;
}

public activity.proxy.types.crmondemand.xml.activity.data.ActivityData
doSerialize2(activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance, XMLWriter writer,
SOAPSerializationContext context) throws Exception
{
    if (instance.getModifiedByld() != null)
    {
        myns3_string__java_lang_String_String_Serializer.setNullable( false );
        myns3_string__java_lang_String_String_Serializer.serialize(instance.getModifiedByld(),
            ns2_ModifiedByld_QNAME, null, writer, context);
    }
    if (instance.getCreatedByld() != null) {
        myns3_string__java_lang_String_String_Serializer.setNullable( false );
        myns3_string__java_lang_String_String_Serializer.serialize(instance.getCreatedByld(),
            ns2_CreatedByld_QNAME, null, writer, context);
    }
    if (instance.getModld() != null) {
        myns3__int__java_lang_Integer_Int_Serializer.setNullable( false );
        myns3__int__java_lang_Integer_Int_Serializer.serialize(instance.getModld(),
            ns2_Modld_QNAME, null, writer, context);
    }
    ...
return instance;
}

```

Entering Telephone Number Values through Web Services

Telephone number values entered through Web service requests and the Oracle CRM On Demand UI are displayed differently both in the UI and in the response to the QueryPage operation. [Table 14](#) shows how telephone number values are displayed in the UI or within the response to a QueryPage request.

Table 14. Formatting of telephone number values entered through Web services or the UI

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

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

Recommended Formats for Telephone Numbers

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

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

- 1 (404) 262-1601 123
- 1 (404) 262-1601 #123
- 1 (140) 426-2160 #1
- 1 (140) 426-2160

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

```
<ListOfContact>
```

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

The following is the query response:

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

Other Considerations for Telephone Numbers

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

+1 (404) 262-1601 #123

is displayed as the following in the Web service response:

+1 404 2621601 #123

and as the following in the UI:

1 (404) 262-1601 #123

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

+1 404 262-1601 #123

is displayed as the following in the Web service response:

1 404 2621601 #123

and as the following in the UI:

1 (404) 262-1601 #123

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

Best Practices for Adhering to Web Service Allotments

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

For the Web Services Operations allotment:

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

For the Web Services Concurrent Request allotment:

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

Working with Record Ownership Modes

For most of the record types that support custom books, the company administrator can set up record ownership for the record type in one of three record ownership modes: user mode, book mode, or mixed mode. If you plan to change ownership modes for record types, you must review and test your Web services integrations to ensure that they continue to work. For more information about record ownership modes, see *Oracle CRM On Demand Online Help*.

Best Practices for Integration Performance

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

Batch Processing

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

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

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

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

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

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

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

Session Management and Pooling

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

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

It is recommended that a user:

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

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

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

The client time-out on a single Web service call must be set to at least 10 minutes, so that the client does not time out when a request is still pending. For information about Web services sessions, see [“About Establishing and Managing the Web Services Session” on page 53](#).

Session Pooling

Session pooling is another option for increasing the performance of your application further. Session pooling involves maintaining a list of active sessions on the client application. The client application must ensure that each session is active and valid (it must have a valid session ID) before using it in a request. The client application can 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 multithreaded application. In a single-threaded application, session pooling can avoid the unnecessary overhead of relogging into Oracle CRM On Demand for each request. In a multithreaded application session, you can use session pooling to run multiple requests at the same time.

API Calls

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

Queries that involve related child objects (that is child objects that are top-level objects), or complex queries that involve criteria from both parent and related child objects, might 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 can 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 might not be apparent at lower volumes, but high volume applications could benefit from the direct update and insert operation, which does not require the additional business logic to determine whether the records already exist.

Working with Attachments

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

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

If you are adding the same attachment to multiple records, you can take advantage of a reuse facility for attachments, see [“Attaching a File to Multiple Records” on page 643](#). In this way, you do not include the content for each record in the request, you use the ContentId on one record and reference that ContentId from other records. For more information about using attachments in Web Services On Demand, see [Appendix A, “Using Attachments with Web Services On Demand.”](#)

Querying Login History and User Usage

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

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

For more information, see [“LoginHistoryQueryPage” on page 178](#) and [“UserUsageQueryPage” on page 206](#).

Using the QueryPage Method

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

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

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

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

For more information about the QueryPage method, see [“QueryPage” on page 91](#), or [“QueryPage” on page 115](#).

Using Indexed Custom Fields

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

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

Indexed custom fields are prefixed with Indexed as a default.

Handling Bad Events in Integration Event Queues

Sometimes an integration event queue might contain an event that the client application cannot process. This situation can occur, for example, if a new custom field has been flagged as required but the client application that processes integration events has not been updated with a schema file that includes the integration tag for the new custom field.

Client applications cannot selectively delete events from the queue, so they must handle these events by performing the following steps:

- 1 In the `GetEvents` call, reduce the number of events retrieved by setting the `EventCount` argument to 1.
- 2 Use `GetEvents` to retrieve single events and process each of them until a bad event is identified.
- 3 If a parse error is encountered, save the bad event to a local folder or other location.
- 4 Delete the bad event from the queue using a `DeleteEvents` call with the `LastEventId` argument set to the ID of the bad event.
- 5 Continue processing events with the number of events set back to the default value.
- 6 Repeat steps 1 through 5 if more bad events are encountered.

For information about `GetEvents` and `DeleteEvents`, see [GetEvents on page 159](#), and [DeleteEvents on page 149](#) respectively.

Best Practices for Integration Management

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

Moving Customers Between Pods

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

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

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

Handling Outages

Outages can be due to:

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

Maintaining SSL Certificates

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

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

6

Web Services On Demand API Calls

This chapter contains the following topics:

- [Web Services On Demand API Calls](#)
- [Web Services v1.0 API Calls on page 84](#)
- [Web Services v2.0 API Calls on page 102](#)
- [Service API Calls on page 139](#)
- [Administrative Services API Calls on page 210](#)
- [Data Loader API Calls on page 280](#)

Web Services On Demand API Calls

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

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

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

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

Web Services v1.0 API Calls

The Web Services v1.0 methods are listed in [Table 15](#). 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 15. Web Services v1.0 Methods

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

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

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

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

User Keys

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

The Echo Argument

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

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

Delete

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

Usage

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

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

NOTE: To conform with Oracle CRM On Demand's business logic, be careful about the order in which objects are deleted. You cannot delete some objects unless some action is performed on its child objects. For example, you cannot delete an account unless you reassociate all its service requests with a different account. For information about the behavior of the Delete method on child objects, see [Table 16](#).

Table 16 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 *Oracle CRM On Demand Online Help*.

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 27.

Table 16. Behavior of Delete Method on Child Objects

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

Table 16. Behavior of Delete Method on Child Objects

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

Arguments

Table 17 describes the arguments taken by the Delete method.

Table 17. Arguments Taken by the Delete Method

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

Return Value of the Call

The status key for each of the deleted objects.

Related Topic

[Update](#).

DeleteChild

Removes child records from the Oracle CRM On Demand database.

Usage

You use the DeleteChild method to remove one or more child records of a particular object from an Oracle CRM On Demand instance, or remove the association between the child and parent object. [Table 16 on page 86](#) is also applicable for the DeleteChild method.

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

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

Arguments

[Table 18](#) describes the arguments taken by the DeleteChild method.

Table 18. Arguments Taken by the DeleteChild Method

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

Return Value of the Call

The status key for each of the deleted child records.

Insert

Inserts new records in the Oracle CRM On Demand database.

Usage

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

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

[Table 19](#) outlines how the Insert method acts on parent and child objects to create or update an object instance.

Table 19. Effect of Insert on Parent and Child Objects

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

Arguments

[Table 20](#) describes the arguments taken by the Insert method.

Table 20. Arguments Taken by the Insert Method

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

Return Value of the Call

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

Related Topic

[Update](#).

InsertChild

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

Usage

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

Arguments

[Table 21](#) describes the arguments taken by the InsertChild method.

Table 21. Arguments Taken by the InsertChild Method

Name	Description	Required	Default	I/O
ListOf(<i>Object</i>). 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 22 outlines how the InsertOrUpdate method acts on parent and child objects to create or update an object instance.

Table 22. Effect of InsetOrUpdate on Parent and Child Objects

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

Arguments

Table 23 describes the arguments taken by the InsertOrUpdate method.

Table 23. Arguments Taken by the InsertOrUpdate Method

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

Return Value of the Call

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

Related Topic

[Update, Insert](#)

QueryPage

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

Query by Template

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

Query by Children

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

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

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

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

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

```
<UseChildAnd>true</UseChildAnd>

<ListOfContact>
  <Contact>
    <ContactFirstName>=' John' </ContactFirstName>
    <ContactLastName/>
    <ListOfAddress>
      <Address>
        <Country>=' Canada' </Country>
      <Address>
    </ListOfAddress>
  </Contact>
</ListOfContact>
```

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

Using the PageSize and StartRowNum Arguments

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

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

```
<StartRowNum>0</StartRowNum>

<PageSi ze>20</PageSi ze>
```

<ListOfAccount>

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

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

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

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

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

About Oracle CRM On Demand Query Syntax

The query syntax supports only a small subset of binary and unary operators. No Siebel Query Language constructs or functions are supported. The query syntax is summarized in [Table 24](#).

Table 24. Query Syntax for QueryPage

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

Table 24. Query Syntax for QueryPage

Syntax Type	Notes
val ue	
' <i>l i t e r a l</i> '	<p>Literal data is always enclosed in single quotes.</p> <p>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.</p> <p>To use the special characters such as asterisk (*), question mark (?), and backslash (\) in queries, precede them with the \ character. For example, to use the ? wildcard operator in a query, precede it with the \ character as follows:</p> <p>\?</p>

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

Querying Oracle CRM On Demand Data Using Web Services

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

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

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

```

abc\*"d
abc\*' "d
(NULL val ue)

```

Table 25 specifies the returned record sets for various values of each field value that maps to the preceding list.

Table 25. Returned Record Sets

Field Value	Returned Record Set	Comments
abc	Not appl i cabl e	An unquoted value without an explicit operator is invalid input.
' abc'	Not appl i cabl e	A quoted value without an explicit operator is invalid input.
= ''' abc	''' ' abc'	None
= "' abc' "	Not appl i cabl e	Double quotes are not allowed by the Oracle CRM On Demand Validator. Consequently, this example returns an error message.
= ' abc	' abc	None
= ' = ' ' abc	''' = ' abc'	None
= ' = ' abc' '	Not appl i cabl e	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
LI KE ' abc\?d	' abc?d	None
LI KE ' abc?d	' abc?d abc*d abc\d	None
~LI KE ' abc?d	' abc?d aBc*D abc*d abc\d	None
= ' abc*d'	abc*d	Any wildcard character that has not been formatted with quotes is treated as if it were formatted with quotes.
= ' abc*d	' abc*d	None
= ' abc*d	' abc*d	None

Table 25. Returned Record Sets

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

Table 25. Returned Record Sets

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

Querying Multiple Fields

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

Example 1: Combining Expressions Using the AND Operator

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

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

Multiple Expressions on a Single Field

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

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

Example 2: Combining Multiple Expressions Using the AND Operator

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

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

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

Example 3: Combining Multiple Expressions Using the OR Operator

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

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

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

Specifying Books in Queries

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

Arguments

Table 26 describes the arguments taken by the QueryPage method.

Table 26. Arguments Taken by the QueryPage Method

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

Return Value of the Call

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

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

Update

Updates records with a new value.

Usage

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

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

Table 27 outlines how the Update method acts on parent and child objects to update an object instance.

Table 27. Effect of Update on Parent and Child Objects

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

Arguments

Table 28 describes the arguments taken by the Update method.

Table 28. Arguments Taken by the Update Method

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

Return Value of the Call

The status key for each of the updated records.

UpdateChild

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

Usage

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

Arguments

[Table 29](#) describes the arguments taken by the UpdateChild method.

Table 29. Arguments Taken by the UpdateChild Method

Name	Description	Required	Default	I/O
ListOf(<i>Object</i>). 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 30](#). These methods can be called on all Oracle CRM On Demand objects (record types), for example, Account, Contact, Opportunity, and so on, including all Custom Objects.

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

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

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

Table 30. Web Services v2.0 API Methods

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

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

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

The Echo Argument

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

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

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

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

The ViewMode Argument

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

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

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

- Personal
- Sales Rep
- Organization

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

The LOVLanguageMode Argument

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

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

For methods other than QueryPage:

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

For the QueryPage method:

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

Delete

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

Usage

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

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

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

Arguments

[Table 31](#) describes the arguments taken by the Delete method.

Table 31. Arguments Taken by the Delete Method

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

Return Value of the Call

The status key for each of the deleted objects.

Sample SOAP Request - Delete

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

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

See [“User Key Fields on the Contact Object” on page 452](#) for a list of valid user key field combinations for the contact object.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

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

      <ListOfContact>

        <Contact>

          <Id>ADSA-96I ZCS</Id>

          <ListOfAccount>

            <Account>

              <AccountName>A1</AccountName>

              <AccountLocation>HQ</AccountLocation>

            </Account>

          </ListOfAccount>

        </Contact>

        <Contact>

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

```

    </Contact>
  </ListOfContact>
</ContactDelete_Input>
</soap: Body>
</soap: Envelope>

```

Sample SOAP Response

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

```

<?xml version="1.0" encoding="utf-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: ContactDelete_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">
        <Contact>
          <ModifiedDate>2011-08-10T15: 21: 43Z</ModifiedDate>
          <CreatedDate>2011-08-09T21: 17: 05Z</CreatedDate>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <CreatedById>HRZ780-I5RUE</CreatedById>
          <ModId>4</ModId>
          <Id>ADSA-96I ZCS</Id>
          <CreatedBy>Joanne Brown, 08/09/2011 16: 17: 05</CreatedBy>
          <ModifiedBy>Joanne Brown, 08/10/2011 10: 21: 43</ModifiedBy>
          <ListOfAccount>
            <Account>
              <ModifiedDate>2011-08-10T14: 44: 27Z</ModifiedDate>
              <CreatedDate>2011-08-10T14: 44: 27Z</CreatedDate>
              <ModifiedById>HRZ780-I5RUE</ModifiedById>
              <CreatedById>HRZ780-I5RUE</CreatedById>
              <ModId>0</ModId>
              <Id>ADSA-977GCL</Id>
              <CreatedBy>Joanne Brown, 08/10/2011 09: 44: 27</CreatedBy>
              <ModifiedBy>Joanne Brown, 08/10/2011 09: 44: 27</ModifiedBy>
            </Account>
          </ListOfAccount>
        </Contact>
      </Contact>
    </Contact>
  </Body>
</Envelope>

```

```
<ModifiedDate>2011-08-10T15:22:46Z</ModifiedDate>
<CreateDate>2011-08-09T21:13:33Z</CreateDate>
<ModifiedById>HRZ780-15RUE</ModifiedById>
<CreatedById>HRZ780-15RUE</CreatedById>
<ModId>2</ModId>
<Id>ADSA-96J5ZL</Id>
<CreatedBy>Joanne Brown, 08/09/2011 16:13:33</CreatedBy>
<ModifiedBy>Joanne Brown, 08/10/2011 10:22:46</ModifiedBy>
</Contact>
</ListOfContact>
</ns:ContactDelete_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Execute

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

Usage

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

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

```
<Account operation="insert">
```

specifies an insert operation for an Account object.

The operation attribute can have one of the following values:

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

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

Arguments

Table 32 describes the arguments taken by the Execute method.

Table 32. Arguments Taken by the Execute Method

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

Return Value of the Call

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

Sample SOAP Request - Execute

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

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

See [“User Key Fields on the Contact Object” on page 452](#) for a list of valid user key field combinations for the contact object.

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

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

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

```
<soap:Header>
```

```

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

```

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

Sample SOAP Response

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

```
<?xml version="1.0" encoding="utf-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:ContactExecute_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">
        <Contact>
          <ModifiedDate>2011-08-25T15:28:34Z</ModifiedDate>
          <CreatedDate>2011-08-25T15:28:34Z</CreatedDate>
          <ModifiedBy>HRZ780-I5RUE</ModifiedBy>
          <CreatedBy>HRZ780-I5RUE</CreatedBy>
          <ModId>0</ModId>
          <Id>ADSA-9HXB9V</Id>
          <CreatedBy>Joanne Brown, 08/25/2011 10:28:34</CreatedBy>
          <ModifiedBy>Joanne Brown, 08/25/2011 10:28:34</ModifiedBy>
        </Contact>
        <Contact>
          <ModifiedDate>2011-08-25T15:28:36Z</ModifiedDate>
          <CreatedDate>2011-08-04T17:01:41Z</CreatedDate>
          <ModifiedBy>HRZ780-I5RUE</ModifiedBy>
          <CreatedBy>HRZ780-I5RUE</CreatedBy>
          <ModId>5</ModId>
          <Id>ADSA-93DZIC</Id>
          <CreatedBy>Joanne Brown, 08/04/2011 12:01:41</CreatedBy>
          <ModifiedBy>Joanne Brown, 08/25/2011 10:28:36</ModifiedBy>
        </Contact>
        <Contact>
          <ModifiedDate>2011-08-12T15:11:17Z</ModifiedDate>
          <CreatedDate>2011-08-12T15:11:17Z</CreatedDate>
          <ModifiedBy>HRZ780-I5RUE</ModifiedBy>
          <CreatedBy>HRZ780-I5RUE</CreatedBy>
          <ModId>0</ModId>
```

```

    <Id>ADSA-98V4V7</Id>

    <CreatedBy>Joanne Brown, 08/12/2011 10:11:17</CreatedBy>

    <ModifiedBy>Joanne Brown, 08/12/2011 10:11:17</ModifiedBy>

  </Contact>

</ListOfContact>

</ns:ContactExecute_Output>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

```

Insert

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

Usage

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

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

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

Arguments

[Table 33](#) describes the arguments taken by the Insert method.

Table 33. Arguments Taken by the Insert Method

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

Table 33. Arguments Taken by the Insert Method

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

Return Value of the Call

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

Sample SOAP Request - Insert

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

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

See [“User Key Fields on the Contact Object” on page 452](#) for a list of valid user key field combinations for the contact object.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

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

      <Echo>0</Echo>

    </ContactInsert_Input>

  </soap:Body>

</soap:Envelope>
```

```

<LOVLanguageMode>LIC</LOVLanguageMode>
<ListOfContact>
  <Contact>
    <ContactFirstName>NewCon2</ContactFirstName>
    <ContactLastName>Con2</ContactLastName>
    <ExternalSystemId>12345009</ExternalSystemId>
  </Contact>
  <Contact>
    <ContactFirstName>Exi stingFi rstName</ContactFi rstName>
    <ContactLastName>Exi sti ngLastName</ContactLastName>
  <ListOfAccount>
    <Account>
      <AccountName>Acti on Rental s</AccountName>
    </Account>
  </ListOfAccount>
</Contact>
</ListOfContact>
</ContactInsert_Input>
</soap: Body>
</soap: Envelope>

```

Sample SOAP Response

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

```

<?xml version="1.0" encoding="utf-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:ContactInsert_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">
        <Contact>
          <ModifiedDate>2011-08-10T14:44:24Z</ModifiedDate>
          <CreateDate>2011-08-10T14:44:24Z</CreateDate>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <CreatedById>HRZ780-I5RUE</CreatedById>
          <ModId>0</ModId>
          <Id>ADSA-977GCJ</Id>
          <CreatedBy>Joanne Brown, 08/10/2011 09:44:24</CreatedBy>

```

```

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

```

QueryPage

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

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

Query by Template

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

Using the `pagesize`, `startrownum`, and `recordcountneeded` Arguments

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

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

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

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

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

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

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

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

Using Search Specifications

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

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

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

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

Query Syntax

The query syntax for the searchspec argument supports only a small subset of binary and unary operators. No Siebel Query Language constructs or functions are supported. The query syntax is summarized in [Table 34](#).

Table 34. Query Syntax for QueryPage

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

Table 34. Query Syntax for QueryPage

Syntax Type	Notes
val ue	
' <i>l i t e r a l</i> '	<p>Literal data is always enclosed in single quotes.</p> <p>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.</p> <p>To use the special characters such as asterisk (*), question mark (?), and backslash (\) in queries, precede them with the \ character. For example, to use the ? wildcard operator in a query, precede it with the \ character as follows:</p> <p>\?</p>

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

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

```
<Contact searchspec=" [ContactFi rstName] = ' 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=" ([ContactFi rstName] = ' Jane' ") AND ([ContactLastName] = ' Doe' ")>
```

Examples of searchspec usage are given in [Table 35](#).

Table 35. Searchspec Examples

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

Table 35. Searchspec Examples

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

Table 35. Searchspec Examples

Operator	Usage of searchspec in request	Description
< (less than)	<pre><ListOfContact><Contact searchspec="[IndexedNumber0] < ' 500' "> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact></pre>	Returns all contacts whose IndexedNumber0 value is less than 500
<= (less than or equal to)	<pre><ListOfContact><Contact searchspec="[IndexedNumber0] <= ' 500' "> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact></pre>	Returns all contacts whose IndexedNumber0 value is less than or equal to 500
<> (not equal to)	<pre><ListOfContact><Contact searchspec="[IndexedNumber0] <> ' 500' "> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact></pre>	Returns all contacts whose IndexedNumber0 value is not equal to 500

The EXISTS and NOT EXISTS Operators

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

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

EXISTS returns true if the field expression matches for at least one of the values of the multivalued field.

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

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

NOT EXISTS returns true if the field expression matches for none of the values of the multivalued field.

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

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

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

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

Specifying the Sort Order and Sort Sequence

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

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

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

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

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

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

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

Examples of `sortorder` and `sortsequence` usage are given in [Table 36](#).

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

CAUTION: Using multiple sort criteria might affect query performance. Use it only when necessary.

Table 36. Sortorder and sortsequence Examples

Sort type	Usage of sortoder and sortsequence	Description
Sort in ascending order	<pre><ListOfContact startrownum="0" page size="100" recordcountneeded="true"> <Contact searchspec="[ContactFi rstName] LIKE 'Contact*' "> <ContactLastName sortorder="ASC"></ ContactLastName> </Contact></Li stOfContact></pre>	Returns all contacts whose <code>ContactFirstName</code> value starts with "Contact" and ends with zero or more characters and sorts them by <code>ContactLastName</code> in ascending order

Table 36. Sortorder and sortsequence Examples

Sort type	Usage of sortorder and sortsequence	Description
Sort in descending order	<pre><ListOfContact startrownum="0" page size="100" recordcountneeded="true"> <Contact> <ContactLastName sortorder="DESC">~LIKE ' Contact*' </ContactLastName> </Contact></ListOfContact></pre>	Returns all contacts whose ContactLastName value starts with "Contact" and ends with zero or more characters and sorts them by ContactLastName in descending order
Sort More than One Field	<pre><ListOfContact startrownum="0" page size="100" recordcountneeded="true"> <Contact searchspec="[ContactFirstName] LIKE ' Contact*' "> <ContactLastName sortorder="DESC" sortsequence="1"></ContactLastName> <ContactFirstName sortorder="DESC" sortsequence="2"></ContactFirstName> <ExternalSystemId sortorder="DESC" sortsequence="3"></ExternalSystemId> </Contact></ListOfContact></pre>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with zero or more characters and sorts them by ContactLastName in descending order and then by ContactFirstName in descending order and then by ExternalSystemId in descending order.

Specifying Books in Queries

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

Querying Lists

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

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

All languages are supported.

Arguments

Table 37 describes the arguments taken by the QueryPage method.

Table 37. Arguments Taken by the QueryPage Method

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

Table 37. Arguments Taken by the QueryPage Method

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

Return Value of the Call

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

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

Sample SOAP Requests

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

Sample SOAP Request 1 - QueryPage with Field-level Filter

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

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <body>

    <QueryPage>

      <pagesize>10</pagesize>

      <startrownum>1</startrownum>

      <recordcountneeded>true</recordcountneeded>

      <ViewMode>Broadest</ViewMode>

      <LOVLanguageMode>LIC</LOVLanguageMode>

      <ContactType>Customer*</ContactType>

    </QueryPage>

  </body>

</soap:Envelope>
```

```

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

```

Sample SOAP Response 1

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

```

<?xml version="1.0" encoding="utf-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: ContactQueryPage_Output xmlns: ns="urn: crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn: /crmondemand/xml/Contact/Data" recordcount="5" lastpage="true">
        <Contact>
          <ContactEmail /></ContactEmail>
          <ContactFirstName>D</ContactFirstName>
          <ContactLastName>Dan</ContactLastName>
          <ContactType>Customer</ContactType>
          <ListOfActivity lastpage="true">

```

```

    <Acti vi ty>
      <Subj ect>Send D Dan wel come email </Subj ect>
    </Acti vi ty>
  </Li stOfActi vi ty>
</Contact>
<Contact>
  <ContactEmail I ></ContactEmail I >
  <ContactFi rstName>L</ContactFi rstName>
  <ContactLastName>Lak</ContactLastName>
  <ContactType>Customer</ContactType>
  <Li stOfActi vi ty l astpage="true">
    <Acti vi ty>
      <Subj ect>Send L Lak wel come email </Subj ect>
    </Acti vi ty>
  </Li stOfActi vi ty>
</Contact>
<Contact>
  <ContactEmail I ></ContactEmail I >
  <ContactFi rstName>NewCon6</ContactFi rstName>
  <ContactLastName>Con6</ContactLastName>
  <ContactType>Customer</ContactType>
  <Li stOfActi vi ty l astpage="true">
    <Acti vi ty>
      <Subj ect>Add acti vi ty to C Contacts</Subj ect>
    </Acti vi ty>
  </Li stOfActi vi ty>
</Contact>
<Contact>
  <ContactEmail I >n@l . com</ContactEmail I >
  <ContactFi rstName>Nathan</ContactFi rstName>
  <ContactLastName>Lak</ContactLastName>
  <ContactType>Customer</ContactType>
  <Li stOfActi vi ty l astpage="true">
    <Acti vi ty>
      <Subj ect>Send Nathan Lak wel come email </Subj ect>
    </Acti vi ty>
  </Li stOfActi vi ty>
</Contact>

```



```
<Acti vi ty>
  <Subj ect>Send Jordan Lak wel come email</Subj ect>
</Acti vi ty>
<Acti vi ty>
  <Subj ect>Send J Lak wel come email</Subj ect>
</Acti vi ty>
<Acti vi ty>
  <Subj ect>Acti vi tyInsert Test R19</Subj ect>
</Acti vi ty>
<Acti vi ty>
  <Subj ect>Acti vi tyInsert Test R19 2</Subj ect>
</Acti vi ty>
<Acti vi ty>
  <Subj ect>Acti vi tyInsert Test R19 2</Subj ect>
</Acti vi ty>
</Li stOfActi vi ty>
</Contact>
<Contact>
  <ContactEmail I></ContactEmail I>
  <ContactFi rstName>NewCon5</ContactFi rstName>
  <ContactLastName>NewCon5</ContactLastName>
  <ContactType>Customer</ContactType>
  <Li stOfActi vi ty Lastpage="true"></Li stOfActi vi ty>
</Contact>
</Li stOfContact>
</ns: ContactQueryPage_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envel ope>
```

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

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

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

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

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

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

Sample SOAP Request 2a

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

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

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

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

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

          <ContactFirstName></ContactFirstName>

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

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

        </Contact>

      </ListOfContact>

    </ContactQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 2a

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

```
<?xml version="1.0" encoding="utf-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:ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" recordcount="54" lastpage="false">
        <Contact>
          <ContactFirstName>James</ContactFirstName>
          <ContactLastName>Conners</ContactLastName>
          <ContactType>Prospect</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Ruth</ContactFirstName>
          <ContactLastName>Chou</ContactLastName>
          <ContactType>Prospect</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Jason</ContactFirstName>
          <ContactLastName>Cheney</ContactLastName>
          <ContactType>Prospect</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Sean</ContactFirstName>
          <ContactLastName>Thomas</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Carlos</ContactFirstName>
          <ContactLastName>Santos</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
      </ListOfContact>
    </ns:ContactQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample SOAP Request 2b

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

```
<?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:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

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

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

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

          <ContactFirstName></ContactFirstName>

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

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

        </Contact>

      </ListOfContact>

    </ContactQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 2b

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

```
<?xml version="1.0" encoding="utf-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:ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">

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

        <Contact>

          <ContactFirstName>Nick</ContactFirstName>

          <ContactLastName>Ray</ContactLastName>

          <ContactType>Customer</ContactType>

        </Contact>

      </ListOfContact>

    </ns:ContactQueryPage_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

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

Sample SOAP Request 3 - QueryPage with NamedSearchSpec

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

[Table 38](#) shows the records in the C Contacts list from the Oracle CRM On Demand UI. There are a total of 11 records in the list, therefore the recordcount value returned in the SOAP response equals 11.

Table 38. Example of Records in a Contact List

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

```

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

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

      <NamedSearchSpec>C Contacts</NamedSearchSpec>

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

        <Contact>

          <ContactFirstName />

          <ContactLastName />

          <ContactType />


```

```

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

```

Sample SOAP Response 3

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

```

<?xml version="1.0" encoding="utf-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: ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" recordcount="11" lastpage="false">
        <Contact>
          <ContactFirstName>David</ContactFirstName>
          <ContactLastName>Cuthbery</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Robin</ContactFirstName>
          <ContactLastName>Coxe</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>John</ContactFirstName>
          <ContactLastName>Connolly</ContactLastName>
          <ContactType>Customer</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>James</ContactFirstName>
          <ContactLastName>Conners</ContactLastName>
          <ContactType>Prospect</ContactType>
        </Contact>
        <Contact>
          <ContactFirstName>Alexander</ContactFirstName>
          <ContactLastName>Cone</ContactLastName>

```

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

Update

Updates the selected record with the new value.

Usage

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

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

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

Arguments

[Table 39](#) describes the arguments taken by the Update method.

Table 39. Arguments Taken by the Update Method

Name	Description	Required	Default	I/O
ListOf(<i>Object</i>). For example, ListOfAccount	The object instance to be updated.	Yes	Not applicable	Input/Output
LOVLanguageMode	The language mode for picklists, see “The LOVLanguageMode Argument” on page 104 .	No	LIC	Input

Table 39. Arguments Taken by the Update Method

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

Return Value of the Call

The status key for the updated objects.

Sample SOAP Request - Update

The following SOAP request updates an existing contact record.

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

See [“User Key Fields on the Contact Object” on page 452](#) for a list of valid user key field combinations for the contact object.

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

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

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

      <Echo>0</Echo>

    </ContactUpdate_Input>

  </soap:Body>

</Envelope>
```

```

<LOVLanguageMode>LIC</LOVLanguageMode>
<ListOfContact>
  <Contact>
    <ContactFirstName>ExistingCon1</ContactFirstName>
    <ContactLastName>ExistingCon1</ContactLastName>
    <ContactEmail>e@e.com</ContactEmail>
  </Contact>
</ListOfContact>
</ContactUpdate_Input>
</soap:Body>
</soap:Envelope>

```

Sample SOAP Response

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

```

<?xml version="1.0" encoding="utf-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:ContactUpdate_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data">
        <Contact>
          <ModifiedDate>2011-08-10T16:42:36Z</ModifiedDate>
          <CreateDate>2011-08-09T21:17:05Z</CreateDate>
          <ModifiedBy>HRZ780-ISRUE</ModifiedBy>
          <CreatedBy>HRZ780-ISRUE</CreatedBy>
          <ModId>8</ModId>
          <Id>ADSA-961ZCS</Id>
          <CreatedBy>Joanne Brown, 08/09/2011 16:17:05</CreatedBy>
          <ModifiedBy>Joanne Brown, 08/10/2011 11:42:36</ModifiedBy>
        </Contact>
      </ListOfContact>
    </ns:ContactUpdate_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Service API Calls

The Oracle CRM On Demand Web Services service methods are listed in [Table 40](#). 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 40. Web Services On Demand Service Methods

Method Name	Web Service	Comments
"AuthAuthorizationUser" on page 140	Authorization	Authenticates the current user and returns information about the user's company and the privileges enabled for the user's role.
"DeletedItemQueryPage" on page 142	Deleted Item	Gets information about deleted items.
"DeleteEvents" on page 149	Integration Event	Deletes events from an integration event queue.
"ExportWebServiceCreateExportRequest" on page 151	Export	Creates an export request.
"ExportWebServiceMonitorExportRequest" on page 154	Export	Gets data about an export request.
"ExportWebServiceRetrieveExportAttachment" on page 157	Export	Gets an attachment for an export request.
"GetEvents" on page 159	Integration Event	Gets events from an integration event queue.
"GetLists" on page 167	Lists	Gets the set of lists for an object.
"GetMapping" on page 170	Mapping Service	Gets a list of the display names of fields for a particular record type and their associated XML tags.
"GetPicklistValues" on page 173	Picklist	Gets lists of picklist values.
"GetServerTime" on page 176	Time	Gets the server time.
"LoginHistoryQueryPage" on page 178	Login History	Gets information about user login history.
"MergeRecords" on page 181	Merge	Merges records.
"MetadataChangeSummaryQueryPage" on page 184	Metadata Change Summary	Gets a summary of changes to metadata.
"SalesProcessQueryPage" on page 187	Sales Process	Gets sales process information.

Table 40. Web Services On Demand Service Methods

Method Name	Web Service	Comments
"SetPassword" on page 193	Password	Sets the passwords of users who use Oracle CRM On Demand.
"SetSessionTimeZone" on page 195	Time	Sets the time zone for a session.
"UpdatePicklist" on page 197	Picklist	Updates picklist values.
"UpdateCascadingPicklists" on page 201	Picklist	Updates cascading picklist values.
"UserUsageQueryPage" on page 206	User Usage	Gets information about Web services utilization.

AuthAuthorizationUser

Authenticates the current user and returns information about the user's company and the list of privileges enabled for the user's role.

Usage

You use the AuthAuthorizationUser method to authenticate the current user and return information relevant to the user.

There are no input argument, the method simply authenticates the current user name and password, and returns the information for the authenticated user.

The optional SOAP header parameter <AuthenticateOnly> is intended for use only with the AuthAuthorizationUser method:

- Setting the <AuthenticateOnly> parameter to true and sending a subsequent request with security credentials results in the previous session being closed (that is, a logoff is performed), and a new session is created with the supplied credentials.
- If the SOAP header parameter <SessionKeepAlive> is set to true, and a subsequent request does not contain security credentials, the previous session remains open and the request will be considered authenticated.

NOTE: If both <AuthenticateOnly> and <SessionKeepAlive> are set to true, <AuthenticateOnly> takes precedence.

Arguments

Table 41 describes the arguments taken by the AuthAuthorizationUser method.

Table 41. Arguments Taken by the AuthAuthorizationUser Method

Name	Description	Required	Default	I/O
AuthenticatedUser	Information about the authenticated user.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned.

- **UserId.** The user ID of the authenticated user.
- **TenantId.** The tenant for the authenticated user (SYS_TENANT_ID).
- **ListofPrivileges.** The set of privileges that have been enabled for the user's role.
- **OrganizationName.** The company name for the user.
- **Location.** The location of the company.
- **RoleId.** The role ID for the user's role.
- **Role.** The user's role.

Sample SOAP Request - AuthAuthorizationUser

The following SOAP request invokes AuthAuthorizationUser. No arguments are required.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <AuthWSAuthAuthorizationUser_Input xmlns="urn:crmondemand/ws/auth/" />

  </soap:Body>

</soap:Envelope>
```

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

Sample SOAP Response

The following illustrates the SOAP response (most of the <Privilege> elements are omitted as indicated by the ellipses (...)):

```
<?xml version="1.0" encoding="utf-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:AuthWS_AuthAuthorizationUser_Output xmlns:ns="urn:crmondemand/ws/auth/">

      <AuthenticatedUser xmlns="urn:/crmondemand/xml/auth">

        <User>

          <UserId>1-CWTEQ</UserId>

          <TenantId>1-CWJ8T</TenantId>

          <OrganizationName>Farley 75</OrganizationName>

          <Location>Toronto</Location>

          <RoleId>1-CWJU5</RoleId>

          <Role>Vertical Admin</Role>

          <ListOfPrivileges>

            <Privilege>

              <PrivilegeName>FULL_ACCESS_ANALYTICS</PrivilegeName>YTICS</PrivilegeName>

            </Privilege>

            ...

            ...

          </ListOfPrivileges>

        </User>

      </AuthenticatedUser>

    </ns:AuthWS_AuthAuthorizationUser_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

DeletedItemQueryPage

Returns details of deleted items.

Objects Supported

DeletedItemQueryPage is supported for all Web Services v1.0 and Web Services v2.0 accessible objects that can be deleted.

Usage

You use the DeletedItemQueryPage method to execute a query against the list of deleted records, and return a subset of the records that match the search criteria set by the method arguments.

The Type of the DeletedItems object returned by the DeletedItemQueryPage method is not always the same as that used in the UI of the Oracle CRM On Demand application, as shown in [Table 42](#).

NOTE: In [Table 42](#), 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 42. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
Account	Account
Address	Address
Contact	Contact
Opportunity	Opportunity
Lead	Lead
Service Request	Service Request
Campaign	Campaign
Appointment	Action***Appointment
Solution	Solution
Account Note	Account Note
Account Private Note	Account Private Note
Contact Note	Contact Note
Contact Private Note	Contact Private Note
Note	Note
Opportunity Note	Opportunity Note
Opportunity Private Note	Opportunity Private Note
Organizations Note	Organizations Note
Service Request Note	Service Request Note
Account Attachment	Account Attachment

Table 42. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
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
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 <i>n</i> (where <i>n</i> = 04 and higher)	CustomObject <i>n</i> (where <i>n</i> = 4 and higher)
Sample Dropped	Pharma Call Sample Dropped
Product Detailed	Pharma Call Product Detailed

Table 42. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
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

Arguments

[Table 43](#) describes the arguments taken by the DeletedItemQueryPage method.

Table 43. 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

Table 43. Arguments Taken by the DeletedItemQueryPage Method

Name	Description	Required	Default	I/O
StartRowNum	Indicates the row from which the DeletedItemQueryPage method starts to return records. Use the StartRowNum argument to return a set of records for any given method. For example, if you want to return records 1-100, you set StartRowNum to 0. Then, if you want to return records 101-200, you set StartRowNum to 100, and run the query again. You continue doing this until the last page is returned. In this way, you can return all records for a particular query.	No	0	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned for deleted items:

- **DeletedItemId**. The ID of the deleted item.
- **DeletedById**. The user ID of the user who deleted the item.
- **DeletedBy**. The name of the user who deleted the item.
- **DeletedDate**. The date on which the item was deleted.
- **Name**. The name of the deleted record.
- **ObjectId**. The object ID of the deleted record.
- **Type**. The type of the deleted record.
- **ExternalSystemId**. The external system ID of the item.

Sample SOAP Request - DeletedItemQueryPage

The following SOAP request queries the list of deleted items with the object type = "Contact". It demonstrates the use of the optional arguments <PageSize> and <StartRowNum> to return a specific number of records starting at a specific row.

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

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

```
<soap:Header>
```

```

<wsse: Security>
  <wsse: UsernameToken>
    <wsse: Username>%%USERNAME%%</wsse: Username>
    <wsse: Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse: Password>
  </wsse: UsernameToken>
</wsse: Security>
</soap: Header>
<soap: Body>
  <DeletedItemWS_DeletedItemQueryPage_Input xmlns="urn:crmondemand/ws/deleteditem/">
    <ListOfDeletedItem>
      <DeletedItem>
        <DeletedItemID />
        <DeletedByID />
        <DeletedBy />
        <DeletedDate />
        <Name />
        <ObjectID />
        <Type>='Contact'</Type>
        <ExternalSystemID />
      </DeletedItem>
    </ListOfDeletedItem>
    <PageSize>3</PageSize>
    <StartRowNum>0</StartRowNum>
  </DeletedItemWS_DeletedItemQueryPage_Input>
</soap: Body>
</soap: Envelope>

```

Sample SOAP Response

The following SOAP response contains three deleted contact records:

```

<?xml version="1.0" encoding="utf-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: DeletedItemWS_DeletedItemQueryPage_Output xmlns:ns="urn:crmondemand/ws/deleteditem/">
      <ns: LastPage>true</ns: LastPage>
      <ListOfDeletedItem xmlns="urn:/crmondemand/xml/deleteditem">

```

```

<DeletedItem>
  <DeletedItemID>ADSA-921PLC</DeletedItemID>
  <DeletedBy>Joanne Brown</DeletedBy>
  <DeletedByID>HRZ780-15RUE</DeletedByID>
  <DeletedDate>08/02/2011 15:19:24</DeletedDate>
  <Name>Person One</Name>
  <ObjectID>ADSA-921PL5</ObjectID>
  <Type>Contact</Type>
  <ExternalSystemID></ExternalSystemID>
</DeletedItem>
<DeletedItem>
  <DeletedItemID>ADSA-93EVM</DeletedItemID>
  <DeletedBy>Joanne Brown</DeletedBy>
  <DeletedByID>HRZ780-15RUE</DeletedByID>
  <DeletedDate>08/04/2011 12:16:39</DeletedDate>
  <Name>Jordan Lak</Name>
  <ObjectID>ADSA-93EGXJ</ObjectID>
  <Type>Contact</Type>
  <ExternalSystemID>ABC321</ExternalSystemID>
</DeletedItem>
<DeletedItem>
  <DeletedItemID>ADSA-93EVFP</DeletedItemID>
  <DeletedBy>Joanne Brown</DeletedBy>
  <DeletedByID>HRZ780-15RUE</DeletedByID>
  <DeletedDate>08/04/2011 12:20:44</DeletedDate>
  <Name>J Lak</Name>
  <ObjectID>ADSA-93ET5I</ObjectID>
  <Type>Contact</Type>
  <ExternalSystemID>abc223</ExternalSystemID>
</DeletedItem>
</ListOfDeletedItem>
</ns:DeletedItemWS_DeletedItemQueryPage_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

DeleteEvents

Deletes events from an integration event queue.

Objects Supported

DeleteEvents is supported for the following objects: Account, Activity, Allocation, Asset, Business Plan, Campaign, Contact, Contact State License, Custom Object 01 - 03, Custom Object 04 and higher, Dealer, Household, Inventory Period, Lead, MedEvent, Objective, Opportunity, Period, Portfolio, Product, ProductCategory, Sample Inventory, Sample Transaction, Sample Lot, Service Request, Solution, Territory, Transaction Item, Vehicle.

Usage

You use the DeleteEvents method of the Integration Event Web service to delete events from an integration event queue. Integration events are actions that are triggered based on meeting certain workflow criteria. An integration event stores information about data that has changed:

- User key information about the changed record, for example: objectID, externalsystemID
- Audit information, for example, created date, createdby, modified date, modified by

Integration events are stored in one or more company queues on the hosted environment. The maximum number of events in the queues is set by Customer Care. Contact Customer Care to request support for the Integration Event Web Service and to specify the maximum total size of the integration queues you require. For more information about integration events and setting up workflow criteria, see *Oracle CRM On Demand Online Help*.

The QueueName argument, if specified, must match the name of a queue defined for a company within Oracle CRM On Demand. The QueueName value is the localized display name value for the user. If the supplied name is not valid, an error message is returned.

If QueueName is not specified, the value is empty, or if the value is Default Queue, events are deleted from the default queue.

If the DateTime argument is supplied, all events older than the specified date and time are deleted. If the LastEventId argument is supplied, all events older than the specified event are deleted. If DateTime and LastEventId are not specified, all events are deleted from the queue.

If a valid queue name is provided but the LastEventId value provided does not match an event in that queue, an error is returned.

When there is a pending request to a queue, another call to the same queue cannot be invoked until the previous request completes. However, another call to a different queue can be invoked before the previous request completes.

You can delete events for all of the supported record types, or a subset of record types, depending on how you prepare the WSDL and Schema files associated with the Integration Event service, see ["Preparing the Integration Event WSDL File and Schema Files" on page 166](#).

Arguments

Table 44 describes the arguments taken by the DeleteEvents method.

Table 44. Arguments Taken by the DeleteEvents Method

Name	Description	Required	Default	I/O
QueueName	The name of the integration event queue.	No	Default Queue	Input/Output
DateTime	A date and time.	No	Not applicable	Input
LastEventId	An event ID	No	Not applicable	Input/Output

Return Value of the Call

The ID of the last event deleted.

Sample SOAP Request - DeleteEvents

The following SOAP request deletes events from the queue specified. If both the <LastEventId> and <DateTime> arguments are supplied, the <LastEventId> is used to determine which events to delete.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <IntegrationEventWS_DeleteEvents_Input xmlns="urn:crmondemand/ws/integrationevent/">

      <DateTime>08/09/2011 10:40:53</DateTime>

      <LastEventId>20110809152116757_QNMHRZ780-I5RT8_Contact_Contact_ADSA-96CRZK_1_ADSA-96CSA5.xml</LastEventId>

      <QueueName>Default Queue</QueueName>

    </IntegrationEventWS_DeleteEvents_Input>

  </soap:Body>

</soap:Envelope>
```

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

Sample Soap Response

The following SOAP response contains the deleted event:

```
<?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_DeleteEvents_Output xmlns:ns="urn:crmondemand/ws/integrationevent">
      <ns:LastEventId>20110809152116757_QNMHRZ780-I5RT8_Contact_Contact_ADSA-96CRZK_1_ADSA-96CSA5.xml </
ns:LastEventId>
      <ns:QueueName>Default Queue</ns:QueueName>
    </ns:IntegrationEventWS_DeleteEvents_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ExportWebServiceCreateExportRequest

Creates a data export request.

Usage

Use the `ExportWebServiceCreateExportRequest` method to create an export request. The `ADMIN_EXPORT` privilege is required.

For more information about export requests, see *Oracle CRM On Demand Online Help*.

Arguments

[Table 45](#) describes the arguments taken by the `ExportWebServiceCreateExportRequest` method.

Table 45. Arguments Taken by the `ExportWebServiceCreateExportRequest` Method

Name	Description	Required	Default	I/O
ListOfRecordType	The record types to be exported.	Yes	Not applicable	Input
TimeZone	The time zone.	Yes	Not applicable	Input

Table 45. Arguments Taken by the ExportWebServiceCreateExportRequest Method

Name	Description	Required	Default	I/O
DateTimeFormat	The date and time format used in the CSV file, for example: USA - MM/DD/YYYY hh:mm:ss PM.	Yes	Not applicable	Input
ExportDataModifiedAfter	Specifies that data modified on or after the given date is exported.	No	Not applicable	Input
ExportDataModifiedBefore	Specifies that data modified before the given date is exported.	No	Not applicable	Input
Delimiter	The type of CSV delimiter used in the file: Comma (,) or Semi-Colon (;).	Yes	Not applicable	Input
RequestId	The export request ID.	Not applicable	Not applicable	Output

Return Value of the Call

The export request ID.

Sample SOAP Requests and Responses

This topic contains sample SOAP requests that invoke ExportWebServiceCreateExportRequest.

Sample SOAP Request 1 - ExportWebServiceCreateExportRequest: Request for a Comma-delimited Export for Multiple Record Types

The following SOAP sample invokes ExportWebServiceCreateExportRequest to request an export of account and contact records. By not specifying a value for <Delimiter>, comma-delimited export files are generated.

```
<?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:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <body>

    <ExportWebServiceCreateExportRequest>

      <ExportDataModifiedAfter>2007-01-01T00:00:00.0000000</ExportDataModifiedAfter>

      <ExportDataModifiedBefore>2007-01-01T00:00:00.0000000</ExportDataModifiedBefore>

      <Delimiter></Delimiter>

      <DateTimeFormat>MM/DD/YYYY hh:mm:ss PM</DateTimeFormat>

      <RequestId></RequestId>

    </ExportWebServiceCreateExportRequest>

  </body>

</soap:Envelope>
```



```

    </wsse: Security>
  </soap: Header>
  <soap: Body>
    <ExportWebService_CreateExportRequest_Input xmlns="urn: crmondemand/ws/
exportwebservice/">
      <TimeZone>(GMT-08:00) Pacific Time (US & Canada); Tijuana</TimeZone>
      <ListOfRecordType>
        <RecordType>
          <Name>Contact</Name>
        </RecordType>
        <RecordType>
          <Name>Account</Name>
        </RecordType>
      </ListOfRecordType>
      <DateTimeFormat>USA - MM/DD/YYYY hh:mm:ss PM</DateTimeFormat>
      <Delimiter></Delimiter>
    </ExportWebService_CreateExportRequest_Input>
  </soap: Body>
</soap: Envelope>

```

Sample SOAP Request 2 - ExportWebServiceCreateExportRequest: Request for a Semicolon-delimited Export of Records Modified Between Two Dates

The following SOAP sample invokes ExportWebServiceCreateExportRequest to request a semicolon-delimited export of contact records modified between 12/06/2011 and 12/07/2011 (6th and 7th December 2011).

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<soap: Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap: Header>
    <wsse: Security>
      <wsse: UsernameToken>
        <wsse: Username>%%USERNAME%%</wsse: Username>
        <wsse: Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText">%%PASSWORD%%</wsse: Password>
      </wsse: UsernameToken>
    </wsse: Security>
  </soap: Header>
  <soap: Body>

```

```
<ExportWebService_ExportWebServiceCreateExportRequest_Input xmlns="urn: crmondemand/ws/
exportwebservice/">

  <ExportDataModifiedBefore>12/07/2011</ExportDataModifiedBefore>

  <TimeZone>(GMT-08:00) Pacific Time (US & Canada); Tijuana</TimeZone>

  <ListOfRecordType>

    <RecordType>

      <Name>Contact</Name>

    </RecordType>

  </ListOfRecordType>

  <DateTimeFormat>USA - MM/DD/YYYY hh:mm:ss PM</DateTimeFormat>

  <ExportDataModifiedAfter>12/06/2011</ExportDataModifiedAfter>

  <Delimiter>Semi-Colon</Delimiter>

</ExportWebService_ExportWebServiceCreateExportRequest_Input>

</soap: Body>

</soap: Envelope>
```

Sample SOAP Responses

The response is the same for both of the sample SOAP requests:

```
<?xml version="1.0" encoding="utf-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: ExportWebService_ExportWebServiceCreateExportRequest_Output xmlns: ns="urn: crmondemand/ws/
exportwebservice/">

      <ns: RequestId>10A2-16438E</ns: RequestId>

    </ns: ExportWebService_ExportWebServiceCreateExportRequest_Output>

  </SOAP-ENV: Body>

</SOAP-ENV: Envelope>
```

ExportWebServiceMonitorExportRequest

Retrieves the status and other details about an export request.

Usage

Use the `ExportWebServiceMonitorExportRequest` method to retrieve the status and other details for an export request. For successfully completed export requests, the `<ListOfAttachmentInfo>` argument contains the set of ID values for the files generated for the request. Export requests that are in progress do not return any values for `<ListOfAttachmentInfo>`.

NOTE: The actual attachment file is not retrieved using this method but must be retrieved using the `ExportWebServiceRetrieveExportAttachment` method.

The `<AttachmentInfo>` child element of `<ListOfAttachmentInfo>` contains the following for each attachment:

- **FileId.** The file ID.
- **FileName.** The filename of the attachment.
- **FileExt.** The file extension.
- **FileSize.** The file size in KB. The maximum file size for an attachment is 20 MB.

Arguments

Table 46 describes the arguments taken by the `ExportWebServiceMonitorExportRequest` method.

Table 46. Arguments Taken by the `ExportWebServiceMonitorExportRequest` Method

Name	Description	Required	Default	I/O
RequestId	The export request ID.	Yes	Not applicable	Input
CompletionDateTime	The date and time of completion of the request.	No	Not applicable	Output
NumberProcessed	The number of records processed by the request.	No	Not applicable	Output
Status	The status of this Web service call.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service call, or a message indicating success.	Not applicable	Not applicable	Output
ListOfAttachmentInfo	The attachment information.	Not applicable	Not applicable	Output

Return Value of the Call

The output arguments.

Sample SOAP Request - `ExportWebServiceMonitorExportRequest`

The following SOAP sample invokes `ExportWebServiceMonitorExportRequest` to retrieve the status and other details of an export request identified by the request ID.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

        <wsse:Username>%%USERNAMER%%</wsse:Username>

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ExportWebServiceExportWebServiceMonitorExportRequest_Input xmlns="urn:crmondemand/ws/exportwebservice/">

      <RequestId>1QA2-16438E</RequestId>

    </ExportWebServiceExportWebServiceMonitorExportRequest_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following SOAP response contains the status and other details of the export request:

```
<?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:ExportWebServiceExportWebServiceMonitorExportRequest_Output xmlns:ns="urn:crmondemand/ws/exportwebservice/">

      <ns:Status>Completed</ns:Status>

      <ns:CompletionDateTime>12/07/2011 08:48:15</ns:CompletionDateTime>

      <ns:NumberProcessed>1</ns:NumberProcessed>

      <ns:ErrorMessage>OK</ns:ErrorMessage>

      <ListOfAttachmentInfo xmlns="http://www.siebel.com/xml/ExportWSAttachmentInfo">

        <AttachmentInfo>

          <FileId>1QA2-166416</FileId>

          <FileName>OnDemandExport_20111207_1QA2-16438E_0001</FileName>

        </AttachmentInfo>

      </ListOfAttachmentInfo>

    </ns:ExportWebServiceExportWebServiceMonitorExportRequest_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

```

    <FileExt>zip</FileExt>
    <FileSize>3385</FileSize>
  </AttachmentInfo>
</ListOfAttachmentInfo>
</ns:ExportWebService_ExportWebServiceMonitorExportRequest_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

ExportWebServiceRetrieveExportAttachment

Retrieves the files generated by an export request.

Usage

Use the `ExportWebServiceRetrieveExportAttachment` method to retrieve an attachment for an export request by specifying the `AttachmentId` value of the attachment. You can retrieve the `AttachmentId` value by using the `ExportWebServiceMonitorExportRequest` method.

The maximum size of an attachment file generated through a Web services call is 20MB. It is possible to generate larger attachment files through the Oracle CRM On Demand UI, but you cannot retrieve such files using the `ExportWebServiceRetrieveExportAttachment` method; such files must be downloaded from the Export Request Queue page in Oracle CRM On Demand.

NOTE: Only one attachment can be retrieved at a time.

Arguments

Table 47 describes the arguments taken by the `ExportWebServiceRetrieveExportAttachment` method.

Table 47. Arguments Taken by the `ExportWebServiceRetrieveExportAttachment` Method

Name	Description	Required	Default	I/O
RequestId	The export request ID.	Yes	Not applicable	Input
AttachmentId	The attachment ID.	Not applicable	Not applicable	Input
Attachment	The attachment retrieved in Base64 encoded format.	Not applicable	Not applicable	Output

Return Value of the Call

The base64-encoded file with the specified Id value is returned.

Sample SOAP Request - `ExportWebServiceRetrieveExportAttachment`

The following SOAP sample invokes `ExportWebServiceRetrieveExportAttachment` to retrieve the export files based on the export request ID and the attachment file ID:

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <ExportWebService_ExportWebServiceRetrieveExportAttachment_Input xmlns="urn:crmondemand/ws/exportwebservice">

      <RequestId>1QA2-16438E</RequestId>

      <AttachmentId>1QA2-166416</AttachmentId>

    </ExportWebService_ExportWebServiceRetrieveExportAttachment_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following SOAP response contains the requested attachment:

```
<?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:ExportWebService_ExportWebServiceRetrieveExportAttachment_Output xmlns:ns="urn:crmondemand/ws/exportwebservice">

      <ns:Attachment>UESDBBQAAAAIAAh0hz/
CCm0j UAEA AHkCAA mAHEARXhwb3JOU3VtbWFyeV8yMDExMTIwN18xUUeYLT E2NDM4RS50eHRTRFwApAAAAAAIAA6Py89j ZGBpEGFgYFBh
gAAHI GZKAj NZFYCEApDNyArhi wKJr2aunmpl cvrK8cnZl HVMDckMLGBpCYb/

      ...

      CCm0j UAEA AHkCAA mAHEAAAAAAAEAI AC2gQAAAABFeHBvcnRTdW1tYXJ5XzIwMTExMTIwN18xUUeYLT E2NDM4RS50eHRTRFwApAAAAAAIAA6Py89j ZGBpEGFgYFBh
AAFPmN90UESBAhcLFAAAAAGACE6HP1i OK00bCwAAkVMAAB0AEQAAAAAAQAgALaBBQIAAENvbnRhY3RzXzFRQTI tMTYOMzhFXzAwMDEuY
3N2U0QEAkQAAABVVAUAB8+Y305QSwUGAAAAAAIAAgDBAAAZA0AAAAA</ns:Attachment>

    </ns:ExportWebService_ExportWebServiceRetrieveExportAttachment_Output>

  </SOAP-ENV:Body>
```

</SOAP-ENV:Envelope>

GetEvents

Returns events from an integration event queue.

Objects Supported

GetEvents is supported for the following objects: Account, Activity, Allocation, Asset, Business Plan, Campaign, Contact, Contact State License, Custom Object 01 - 03, Custom Object 04 and higher, Dealer, Household, Inventory Period, Lead, MedEvent, Objective, Opportunity, Period, Portfolio, Product, ProductCategory, Sample Inventory, Sample Transaction, Sample Lot, Service Request, Solution, Territory, Transaction Item, Vehicle.

Usage

You use the GetEvents method of the Integration Event Web service to return events from an integration event queue. Integration events are actions that are triggered based on meeting certain workflow criteria. An integration event stores information about data that has changed:

- User key information about the changed record, for example: objectID, externalsystemID
- Audit information, for example, created date, created by, modified date, modified by

For integration events resulting from merge operations, the objectID and externalsystemID of both the primary and duplicate records are returned:

- DeletedRecordExternalSystemId
- DeletedRecordId
- MergedRecordExternalSystemId
- MergedRecordId

Integration events are stored in one or more company queues on the hosted environment. The maximum number of events in the queues is set by Customer Care. Contact Customer Care to request support for the Integration Event Web Service and to specify the maximum total size of the integration queues you require.

For more information about integration events and setting up workflow criteria, see *Oracle CRM On Demand Online Help*.

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

The QueueName argument, if specified, must match the name of a queue defined for a company within Oracle CRM On Demand. The QueueName value is the localized display name value for the user. If the supplied name is not valid, an error message is returned.

If QueueName is not specified, the value is empty, or if the value is Default Queue, events are returned from the default queue.

When there is a pending request to a queue, another call to the same queue cannot be invoked until the previous request completes. However, another call to a different queue can be invoked before the previous request completes.

You can return events for all supported record types, or a subset of record types, depending on how you prepare the WSDL and Schema files associated with the Integration Event service, see [“Preparing the Integration Event WSDL File and Schema Files” on page 166](#).

In some cases the names of objects in the list of events returned differ from the name of the object used in the UI of Oracle CRM On Demand, as shown in [Appendix B, “Mapping of Display Names to Service API Object Names.”](#)

Arguments

[Table 48](#) describes the arguments taken by the GetEvents method.

Table 48. Arguments Taken by the GetEvents Method

Name	Description	Required	Default	I/O
EventCount	The maximum number of events to be returned.	No	Not applicable	Input
QueueName	The integration event queue name.	No	Default queue	Input/ Output
ListOfEvent	A list of events	Not applicable	Not applicable	Output
LastEventID	An event ID	Not applicable	Not applicable	Output

Return Value of the Call

A list of the events returned from the integration event queue. Also, the ID of the last event returned.

The QueueName value is returned, if it is specified in the GetEvents request.

For each event in the list of events, there are the following attributes:

- **name.** The name of the associated Workflow.
- **object.** The record type.
- **operation.** The operation performed. The attribute values can be:
 - insert - for items inserted
 - update - for items updated
 - merge - for items merged
 - 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
 - restore - for items that have been deleted and then restored
 - 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 Activity, 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 *Oracle CRM On Demand Online Help*.

Sample SOAP Request and Responses

This topic contains sample SOAP requests that invoke GetEvents.

Sample SOAP Request 1 - GetEvents with No Arguments

In the following SOAP request, no arguments are passed in (there are no required arguments for GetEvents), all the events from the default queue are retrieved:

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <IntegrationEventWS_GetEvents_Input xmlns="urn:crmondemand/ws/integrationevent/"></IntegrationEventWS_GetEvents_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response 1

The following shows the response for sample request 1. The events from the default queue are returned.

```
<?xml version="1.0" encoding="utf-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 object="Contact" name="cContact Update" operation="update">
    <Sibel Message>
      <ListOfContact xmlns="urn: /crmondemand/xml /contact">
        <Contact>
          <ContactId>ADSA-96CRZK</ContactId>
          <ModId>1</ModId>
          <CreateDate>08/09/2011 10:17:48</CreateDate>
          <ExternalSystemId></ExternalSystemId>
          <ContactFirstName>Testing</ContactFirstName>
          <IntegrationId>ADSA-96CRZK</IntegrationId>
          <ContactLastName>U2</ContactLastName>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <ModifiedDate>08/09/2011 10:21:16</ModifiedDate>
          <AlternateAddressId>ADSA-6CRZM</AlternateAddressId>
          <CreatedById>HRZ780-I5RUE</CreatedById>
        </Contact>
      </ListOfContact>
    </Sibel Message>
  </Event>
  <Event object="Contact" name="cContact - Send Welcome Package" operation="insert">
    <Sibel Message>
      <ListOfContact xmlns="urn: /crmondemand/xml /contact">
        <Contact>
          <ContactId>ADSA-96D7A0</ContactId>
          <ModId>0</ModId>
          <AccountId>No Match Row Id</AccountId>
          <CreateDate>08/09/2011 10:40:52</CreateDate>
          <ExternalSystemId></ExternalSystemId>
          <ContactFirstName>Testing2</ContactFirstName>
          <IntegrationId>ADSA-96D7A0</IntegrationId>
          <ContactLastName>U1</ContactLastName>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <ModifiedDate>08/09/2011 10:40:53</ModifiedDate>
          <AlternateAddressId>ADSA-96D7A0</AlternateAddressId>
          <CreatedById>HRZ780-I5RUE</CreatedById>
        </Contact>
      </ListOfContact>
    </Sibel Message>
  </Event>
</ListOfEvent>

```

```

    </Contact>
  </ListOfContact>
</Siebel Message>
</Event>
<Event object="Contact" name="cContact Update" operation="update">
  <Siebel Message>
    <ListOfContact xmlns="urn: /crmondemand/xml /contact">
      <Contact>
        <ContactId>ADSA-96D7EA</ContactId>
        <ModId>2</ModId>
        <CreateDate>08/09/2011 10: 45: 39</CreateDate>
        <External SystemId></External SystemId>
        <ContactFirstName>C</ContactFirstName>
        <Integrati onId>ADSA-96D7EA</Integrati onId>
        <ContactLastName>C8</ContactLastName>
        <Modi fi edById>HRZ780-I 5RUE</Modi fi edById>
        <Modi fi edDate>08/09/2011 10: 57: 52</Modi fi edDate>
        <AI ternateAddressId>ADSA-96D7EC</AI ternateAddressId>
        <CreatedById>HRZ780-I 5RUE</CreatedById>
      </Contact>
    </ListOfContact>
  </Siebel Message>
</Event>
<Event object="Account" name="Accounts Create" operation="insert">
  <Siebel Message>
    <ListOfAccount xmlns="urn: /crmondemand/xml /account">
      <Account>
        <AccountId>ADSA-96DJL4</AccountId>
        <ModId>0</ModId>
        <CreateDate>08/09/2011 11: 04: 50</CreateDate>
        <CurrencyCode>USD</CurrencyCode>
        <External SystemId></External SystemId>
        <Integrati onId>ADSA-96DJL4</Integrati onId>
        <Locati on></Locati on>
        <Modi fi edById>HRZ780-I 5RUE</Modi fi edById>
        <Modi fi edDate>08/09/2011 11: 04: 51</Modi fi edDate>
      </Account>
    </ListOfAccount>
  </Siebel Message>
</Event>

```

```

        <AccountName>A1</AccountName>

        <NumberEmployees>1000</NumberEmployees>

        <PrimaryBillToAddressId>ADSA-96DJL6</PrimaryBillToAddressId>

        <PrimaryShipToAddressId>ADSA-96DJL8</PrimaryShipToAddressId>

        <CreatedBy>HRZ780-15RUE</CreatedBy>

    </Account>
</ListOfAccount>
</SiebelMessage>
</Event>
</ListOfEvent>
<ns:LastEventId>20110809160454038_QNMHRZ780-15RT8_Account_Account_ADSA-96DJL4_0_ADSA-96DCMZ.xml</ns:LastEventId>
</ns:IntegrationEventWS_GetEvents_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Sample SOAP Request 2 - GetEvents with EventCount and QueueName

The following SOAP request retrieves one event from the queue High Priority Queue specified by using the <EventCount> and <QueueName> arguments.

```

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

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

    <soap:Header>

        <wsse:Security>

            <wsse:UsernameToken>

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

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

            </wsse:UsernameToken>

        </wsse:Security>

    </soap:Header>

    <soap:Body>

        <IntegrationEventWS_GetEvents_Input xmlns="urn:crmondemand/ws/integrationevent/">

            <QueueName>High Priority Queue</QueueName>

            <EventCount>1</EventCount>

        </IntegrationEventWS_GetEvents_Input>

    </soap:Body>

</soap:Envelope>

```

Sample SOAP Response 2

The following shows the response for sample SOAP request 2. The events from the named queue are returned.

```
<?xml version="1.0" encoding="utf-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 object="Contact" name="cContact Update" operation="update">

          <Siebel Message>

            <ListOfContact xmlns="urn:/crmondemand/xml/contact">

              <Contact>

                <ContactId>ADSA-96CRZK</ContactId>

                <ModId>1</ModId>

                <CreateDate>08/09/2011 10:17:48</CreateDate>

                <ExternalSystemId></ExternalSystemId>

                <ContactFirstName>Testing</ContactFirstName>

                <IntegrationId>ADSA-96CRZK</IntegrationId>

                <ContactLastName>U2</ContactLastName>

                <ModifiedById>HRZ780-I5RUE</ModifiedById>

                <ModifiedDate>08/09/2011 10:21:16</ModifiedDate>

                <AlternateAddressId>ADSA-96CRZM</AlternateAddressId>

                <CreatedById>HRZ780-I5RUE</CreatedById>

              </Contact>

            </ListOfContact>

          </Siebel Message>

        </Event>

      </ListOfEvent>

      <ns:LastEventId>20110809152116757_QNMHRZ780-I5RT8_Contact_Contact_ADSA-96CRZK_1_ADSA-96CSA5.xml</ns:LastEventId>

      <ns:QueueName>Sample Queue</ns:QueueName>

    </ns:IntegrationEventWS_GetEvents_Output>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

Preparing the Integration Event WSDL File and Schema Files

You can use the methods of the Integration Events service to track changes for the supported record types. You must download the `integrationevent.wsdl` file and the XSD file for each record type that you require to track. The integration event WSDL file imports the XSD files for each record type.

In addition, to get integration events generated for merge operations, you must download a Merge schema file and integrate this and the integration event WSDL file with your application. Existing customers must download the latest `integrationevent.wsdl` file, if they wish to use the Merge schema file.

NOTE: The Merge schema file is only applicable to integration events processing and cannot be used for other purposes.

The `integrationevent.wsdl` file is dynamically generated based on which objects have an active workflow rule with an integration event action configured.

For each object with an integration event action configured, a child element of `<SiebelMessage>` is added to the WSDL files, for example, `<ListOfAccount>`.

For child objects, an element is added for the parent object, for example, an entry for the Account parent object is added for the Account Team child. For child objects with multiple parents, an element is added for each possible parent, for example, Account, Contact, and Opportunity parent entries would be added for a Revenue child.

Also, for each object, an `<import>` element is added to the WSDL file with the correct namespace and schema location for the XSD file for the object. For example:

```
<xsd:import namespace="urn:/crmondemand/xml/account" schemaLocation="Account.xsd" />
```

Either the Web Services v1.0 or Web Services 2.0 namespace is used, depending on the object. You can download XSD Schema files for all objects for the Web Services v1.0 API, and for the Web Services v2.0 objects for which Integration Events support is offered. The correct namespace is generated in the WSDL file accordingly.

To prepare the WSDL

- 1 Go to the Web Services Administration page in the Oracle CRM On Demand application.
- 2 Download the Integration Events WSDL.
- 3 Download the Generic Schema files for objects for which you wish to generate integration events and copy these to the folder where you downloaded the WSDL file.

NOTE: You can download Custom or Generic Schema files for all objects for the Web Services v1.0 API, and for the Web Services v2.0 API objects for which Integration Events support is offered.

- 4 If you have created custom fields or renamed fields for an object, download the Custom XSD file for the object.
- 5 If you require to access events resulting from merge operations, download the generic schema file for the Merge Service API.
- 6 In the `integrationevent.wsdl` file, remove references to any Schema files for objects you will not be using.

7 Add the `integrationevent.wsdl` file to your development environment.

NOTE: You must repeat steps 3 through 7 each time a new custom field is added that will be tracked through integration events, or when you begin tracking a new object type. Otherwise, the client application cannot process events containing these newly added fields or objects.

GetLists

Gets the set of lists that the current user has access to for a specified object.

Usage

You use the `GetLists` method to query the details of the lists for an object.

The `ObjectName` argument specifies the object to be queried. The names of public and private lists to which the user has access is returned. The set of list names is equivalent to the set of list names visible to the user in the Manage Lists page UI.

`ListQueryPage` is supported for all objects that support list management.

The list names returned can be used in a `QueryPage` operation to retrieve the set of records matching the list filter criteria.

Arguments

Table 49 describes the arguments taken by the `GetLists` method.

Table 49. Arguments Taken by the `GetLists` Method

Name	Description	Required	Default	I/O
<code>ObjectName</code>	The name of the object.	Yes	Not applicable	Input
<code>ListOfLists</code>	The list of lists returned for the object.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned for each list.

- **Name.** The name of the list.
- **ListType.** Whether the list is a public list, or a private list created by the current user.

Sample SOAP Request - `GetLists`

The following SOAP request retrieves the set of lists visible to the authenticated user for the account record type as specified in the `<ObjectName>` argument:

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

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%USERNAME%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <ListsWS_GetLists_Input xmlns="urn:crmondemand/ws/Lists/">
      <ObjectName>Account</ObjectName>
    </ListsWS_GetLists_Input>
  </soap:Body>
</soap:Envelope>
```

Sample SOAP Response

The following response contains the lists for the account object:

```
<?xml version="1.0" encoding="utf-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:ListsWS_GetLists_Output xmlns:ns="urn:crmondemand/ws/Lists/">
      <ListOfLists xmlns="urn:/crmondemand/xml/Lists">
        <List>
          <Name>All Account Competitor Accounts</Name>
          <ListType>System</ListType>
        </List>
        <List>
          <Name>All Account Partner Accounts</Name>
          <ListType>System</ListType>
        </List>
        <List>
          <Name>All Accounts</Name>
          <ListType>System</ListType>
        </List>
      </ListOfLists>
    </ns:ListsWS_GetLists_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```
</List>
<List>
  <Name>All Customer Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>All Prospect Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>All Referenceable Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>All Top Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>My Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>My Recently Created Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>My Recently Modified Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>Recently Created Accounts</Name>
  <ListType>System</ListType>
</List>
<List>
  <Name>Recently Modified Accounts</Name>
  <ListType>System</ListType>
```

```
</List>
</ListOfLists>
</ns:ListsWS_GetLists_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetMapping

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

Objects Supported

GetMapping is supported for all objects accessible through Web Services V1.0 and Web Services 2.0 and all of their child components.

Usage

You use the GetMapping method to return the mapping between the display names of all the fields in a particular record type and the XML tags for each field.

In some cases, the names of objects used in the `ObjectName` argument differ from the name of the object used in the UI of Oracle CRM On Demand, as shown in [Appendix B, “Mapping of Display Names to Service API Object Names.”](#)

The GetMapping method works with all Custom Objects. However, you must use a different naming convention for Custom Objects 1-3, compared to Custom Object 4 and higher, where there are no spaces in the object name. SOAP requests must use the following naming convention for object names:

- **Custom Object 1-3.** Custom Object 1, Custom Object 2, Custom Object 3
- **Custom Object 4 and higher.** CustomObject4...CustomObject14, and so on

For some fields, the `DisplayName` returned has the value `hidden`. These fields are available in the WSDL file for the object, but are not available in the Oracle CRM On Demand UI. Examples of fields for which the value `hidden` is returned include the `LanguageCode`, `LocaleCode`, `RoleId`, and `TimeZoneId` fields of the User object.

Arguments

[Table 50](#) describes the arguments taken by the GetMapping method.

Table 50. Arguments Taken by the GetMapping Method

Name	Description	Required	Default	I/O
ObjectName	The name of the record type for which you wish to return a list of mappings.	Yes	Not applicable	Input/Output

Return Value of the Call

A list of the display names for fields and their associated XML mappings:

- **LastUpdated.** The date the field was last updated.
- **DisplayName.** The display name of the field, in the user's language. For some fields not available in the UI, the value hidden is returned.
- **ElementName.** The XML element name for the field.
- **DataType.** The field type of the field, for example, Check box, Picklist, and so on.

Sample SOAP Request - GetMapping

The following SOAP request retrieves the display names and XML element names of the fields of the book record type as specified in the <ObjectName> argument:

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <MappingWS_GetMapping_Input xmlns="urn:crmondemand/ws/mapping/">

      <ObjectName>Book</ObjectName>

    </MappingWS_GetMapping_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains the mappings for a number of fields for the book object:

```
<?xml version="1.0" encoding="utf-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:MappingWS_GetMapping_Output xmlns:ns="urn:crmondemand/ws/mapping/">
```

```

<ns:ObjectName>Book</ns:ObjectName>
<ListOfField xmlns="urn:/crmondemand/xml/mappingservice">
  <Field>
    <DisplayName>Can Contain Data</DisplayName>
    <ElementName>CanContainDataFlag</ElementName>
    <DataType>Checkbox</DataType>
  </Field>
  <Field>
    <DisplayName>Id</DisplayName>
    <ElementName>BookId</ElementName>
    <DataType>ID</DataType>
  </Field>
  <Field>
    <DisplayName>Parent Book</DisplayName>
    <ElementName>ParentBookName</ElementName>
    <DataType>Picklist</DataType>
  </Field>
  <Field>
    <DisplayName>Created</DisplayName>
    <ElementName>CreatedBy</ElementName>
    <DataType>Text (Long)</DataType>
  </Field>
  <Field>
    <DisplayName>Book Type</DisplayName>
    <ElementName>BookType</ElementName>
    <DataType>Picklist</DataType>
  </Field>
  <Field>
    <DisplayName>Modified: Date</DisplayName>
    <ElementName>ModifiedDate</ElementName>
    <DataType>Date/Time</DataType>
  </Field>
  <Field>
    <DisplayName>Created: Date</DisplayName>
    <ElementName>CreatedDate</ElementName>
    <DataType>Date/Time</DataType>
  </Field>

```

```

    </Field>
  <Field>
    <DisplayName>Description</DisplayName>
    <ElementName>Description</ElementName>
    <DataType>Text (Long)</DataType>
  </Field>
  <Field>
    <DisplayName>Book Name</DisplayName>
    <ElementName>BookName</ElementName>
    <DataType>Text (Short)</DataType>
  </Field>
</ListOfField>
</ns: MappingWS_GetMapping_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

GetPicklistValues

Gets picklist values from Oracle CRM On Demand.

Objects Supported

GetPicklistValues is supported for all Web Services v1.0 and v2.0 accessible parent-level objects.

Usage

You use the GetPicklistValues method to enable external applications to present lists of values to users, typically in a language-dependent manner. The method can get lists of possible values for both cascading and regular picklist fields.

Because On Demand Web Services is language-independent, it is the client application's responsibility to convert code from the language-independent code (LIC) used by Oracle CRM On Demand to language-dependent values (LDVs) typically used by the external presentation layer.

The returned list of values corresponds to the organization to which the current user belongs (that is, the user whose credentials have been passed during the log-in call).

Cascading picklists restrict the values of one picklist, the related picklist, based on the value selected in another picklist, the parent picklist. For example, a parent picklist might present a list of IT areas and drive the value of a related picklist called SubAreas. When the user selects, for example, the value Installation for Area, the SubAreas picklist is dynamically constrained to show only the picklist values that are associated with the Installation area, for example, Server Crash and No Admin Login.

If the provided picklist has a parent, only the values that have a parent are returned. When a picklist has a parent, the result set includes the parent and the child values and at the end includes an empty set that contains all values available for the requested picklist.

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

- **ParentFieldName**
- **ParentDisplayValue**
- **ParentCode**

If a "10/2004" namespace is used, the **FieldName** and **ParentFieldName** elements respectively accept and return the integration tag value for custom fields, otherwise, they accept and return the generic custom field tag names (that is, **CustomPicklist1** and so on).

Arguments

Table 51 describes the arguments taken by the **GetPicklistValues** method.

Table 51. 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	<i>User's Default Language</i>	Input
ListOfParentPicklistValue	A sequence of ParentPicklistValue elements.	Yes	Not applicable	Output

Return Value of the Call

A list of picklist values. For a cascading picklist, this includes the values for the related picklist that apply for particular values of the parent picklist. For a regular picklist, values for parent picklist are not included.

The **ParentPicklistValue** element contains the following child elements:

- **Language**. The language.
- **ParentFieldName**. The parent picklist field name as an integration tag.
- **ParentDisplayValue**. A display value translated into the specified language.
- **ParentCode**. A parent Language Independent Code (LIC).

- **ListOfPicklistValue.** A sequence of PicklistValue elements containing the related picklist values that correspond to the parent picklist value.

The PicklistValue element contains the following child elements:

- **DisplayValue.** The display value translated into the specified language.
- **Code.** The Language Independent Code (LIC).

Sample SOAP Request - GetPicklist

The following SOAP request retrieves the picklist values in French for the Priority picklist of the account record type.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <PicklistWS_GetPicklistValues_Input xmlns="urn:crmondemand/ws/picklist/">

      <RecordType>Account</RecordType>

      <FieldName>Priority</FieldName>

      <LanguageCode>FRA</LanguageCode>

    </PicklistWS_GetPicklistValues_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains values for the Priority picklist of Account, including the French language display values:

```
<?xml version="1.0" encoding="utf-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: PicklistWS_GetPicklistValues_Output xmlns:ns="urn: crmondemand/ws/picklist/">
  <ListOfParentPicklistValue xmlns="urn: /crmondemand/xml /picklist">
    <ParentPicklistValue>
      <Language>FRA</Language>
      <ParentFieldname></ParentFieldname>
      <ParentDisplayValue></ParentDisplayValue>
      <ParentCode></ParentCode>
      <Displayed></Displayed>
      <ListOfPicklistValue>
        <PicklistValue>
          <Code>Low</Code>
          <DisplayValue>Faible</DisplayValue>
          <Displayed>N</Displayed>
        </PicklistValue>
        <PicklistValue>
          <Code>Medium</Code>
          <DisplayValue>Moyen</DisplayValue>
          <Displayed>N</Displayed>
        </PicklistValue>
        <PicklistValue>
          <Code>High</Code>
          <DisplayValue>El évé</DisplayValue>
          <Displayed>N</Displayed>
        </PicklistValue>
      </ListOfPicklistValue>
    </ParentPicklistValue>
  </ListOfParentPicklistValue>
</ns: PicklistWS_GetPicklistValues_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

GetServerTime

Returns the time from a server.

Usage

You use the `GetServerTime` method to get the time at the server involved in a Web services API session. The time returned is converted to the time for the locale of the user ID making the request.

Return Value of the Call

The current server time.

Sample SOAP Request - `GetServerTime`

The following SOAP request retrieves the server time. No arguments are required.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <TimeWS_GetServerTime_Input xmlns="urn:crmondemand/ws/time/"></TimeWS_GetServerTime_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains the time:

```
<?xml version="1.0" encoding="utf-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:TimeWS_GetServerTime_Output xmlns:ns="urn:crmondemand/ws/time/">

      <ns:CurrentServerTime>08/05/2011 10:54:27</ns:CurrentServerTime>

      <ns:TimeZone>(GMT-06:00) Central Time (US & Canada)</ns:TimeZone>

    </ns:TimeWS_GetServerTime_Output>

  </SOAP-ENV:Body>
```

</SOAP-ENV:Envelope>

LoginHistoryQueryPage

Executes a query against the list of user login history, and returns a subset of the records that match the search criteria set by the method arguments.

Usage

You use the LoginHistoryQueryPage method to view the Login History for a user. This is the same information that is displayed in the Company Administration, Sign In Audit page in the Oracle CRM On Demand application. As an example of how you might use this data, you might save the data in a CSV file and then import it into a spreadsheet. You could then use the spreadsheet to generate a report showing, for example, how often a user logs into Oracle CRM On Demand.

Arguments

[Table 52](#) describes the arguments taken by the LoginHistoryQueryPage method.

Table 52. 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 52. 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.

Sample SOAP Request - LoginHistoryQueryPage

The following SOAP request queries against the list of user login history to return a login record that occurred after the timestamp of 08/01/2011 00:00:00. It also demonstrates the use of the optional <PageSize> and <StartRowNum> arguments to return one record at row 1 (second record).

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

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

```
<soap:Header>
```

```
<wsse:Security>
```

```
<wsse:UsernameToken>
```

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

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

```
</wsse:UsernameToken>
```

```
</wsse:Security>
```

```
</soap:Header>
```

```
<soap:Body>
```

```
<Logi nHi storyWS_Logi nHi storyQueryPage_I nput xmlns="urn:crmondemand/ws/Logi nHi story/">
```

```
<Li stOfLogi nHi story>
```

```
<Logi nHi story>
```

```
<UserI d />
```

```
<Fi rstName />
```

```
<Last Name />
```

```
<UserAl i as />
```

```
<Logi nName />
```

```
<Logi nStatus />
```

```
<Logi nTi mestamp>&gt;=' 08/01/2011 00:00:00'</Logi nTi mestamp>
```

```
<Cl i entType />
```

```
<I PAddress />
```

```
<Addi ti onal I nformati on />
```

```
</Logi nHi story>
```

```
</Li stOfLogi nHi story>
```

```
<PageSi ze>1</PageSi ze>
```

```
<StartRowNum>1</StartRowNum>
```

```
</Logi nHi storyWS_Logi nHi storyQueryPage_I nput>
```

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

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

Sample SOAP Response

The following response contains the requested login history:

```
<?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:Logi nHi storyWS_Logi nHi storyQueryPage_Output xmlns:ns="urn:crmondemand/ws/Logi nhi story/">
      <ns:LastPage>false</ns:LastPage>
      <ListOfLogi nHi story xmlns="urn:/crmondemand/xml/Logi nhi story">
        <Logi nHi story>
          <UserId>HRZ780-15RUE</UserId>
          <FirstName>Joanne</FirstName>
          <LastName>Brown</LastName>
          <UserAlias>Joanne Brown</UserAlias>
          <Logi nName>%%%USERNAME%%%</Logi nName>
          <Logi nStatus>Success</Logi nStatus>
          <Logi nTimestamp>08/02/2011 13:53:00</Logi nTimestamp>
          <ClientType>WS</ClientType>
          <IPAddress>10.156.87.45</IPAddress>
          <AdditionalInformation></AdditionalInformation>
        </Logi nHi story>
      </ListOfLogi nHi story>
    </ns:Logi nHi storyWS_Logi nHi storyQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

MergeRecords

Merges records for certain record types.

Objects Supported

MergeRecords is only supported for the Account, Contact, Household, Lead, and Portfolio record types. If an invalid record type is provided, an error message is displayed.

Usage

You use the MergeRecords method to merge records. When you merge two records, you specify the record that you want to keep, which is called the *primary record*, and the record that is to be deleted, which is called the *duplicate record*.

You must identify a reference to a primary record and a reference to a duplicate record in the request. To identify the primary record, you must specify either a `PrimaryId` or a `PrimaryExternalSystemId` argument. To identify the duplicate record, you must specify either a `DuplicateId` or a `DuplicateExternalSystemId` argument.

The following rules apply to merging records:

- Fields in the primary parent record that contain data are retained.
- Fields in the primary record that are blank get the value from the duplicate record, if it has a value and if the `MergeWhenPrimaryBlank` argument is set to true.
- Fields in the primary parent record that are blank remain blank, if the `MergeWhenPrimaryBlank` argument is not set, or is set to a value other than true.

The Merge Web service has the same security restrictions as in the Oracle CRM On Demand UI regarding privilege and record permissions.

Table 53 describes the arguments taken by `MergeRecords`.

Table 53. Arguments taken by `MergeRecords`

Field Name	Description	Required	Default	I/O
<code>PrimaryId</code>	The ID of the primary record.	One of these arguments is required	Not applicable	Input
<code>PrimaryExternalSystemId</code>	The <code>externalsystemID</code> of the primary record.		Not applicable	Input
<code>DuplicateId</code>	The ID of the duplicate record.	One of these arguments is required	Not applicable	Input
<code>DuplicateExternalSystemId</code>	The <code>externalsystemID</code> of the duplicate record.		Not applicable	Input
<code>MergeWhenPrimaryBlank</code>	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
<code>RecordType</code>	The record type; this is case sensitive.	Yes	Not applicable	Input

Return Value of the Call

The following four values are returned:

- **MergedRecordId.** The ID of the merged record, that is, the primary record.

- **MergedRecordExternalId.** The externalSystemId of the merged record.
- **DeletedRecordId.** The ID of the deleted record, that is, the duplicate record.
- **DeletedRecordExternalId.** The externalSystemId of the merged record.

Sample SOAP Request - MergeRecords

The following SOAP request merges a contact record with DuplicateExternalSystemId = abc223 into a contact record with Id = ADSA-93DZIC.

NOTE: To identify the primary record, you can specify either the `<PrimaryId>` or `<PrimaryExternalSystemId>` argument, and to specify the duplicate record you can use either the `<DuplicateId>` or `<DuplicateExternalSystemId>` argument.

The sample also demonstrates passing in the optional `<MergeWhenPrimaryBlank>` argument to copy the value for a field in the duplicate record to the primary record when the field in the primary record is blank.

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <MergeRecordsWS_MergeRecords_Input xmlns="urn:crmondemand/ws/mergerecords/">

      <PrimaryId>ADSA-93DZIC</PrimaryId>

      <DuplicateExternalSystemId>abc223</DuplicateExternalSystemId>

      <MergeWhenPrimaryBlank>Y</MergeWhenPrimaryBlank>

      <RecordType>Contact</RecordType>

    </MergeRecordsWS_MergeRecords_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response contains the IDs of the merged and deleted records:

```

<?xml version="1.0" encoding="utf-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: DeletedRecordExternal SystemId>abc223</ns: DeletedRecordExternal SystemId>

      <ns: DeletedRecordId>ADSA-93ET5I</ns: DeletedRecordId>

      <ns: MergedRecordExternal SystemId>ABC123</ns: MergedRecordExternal SystemId>

      <ns: MergedRecordId>ADSA-93DZIC</ns: MergedRecordId>

    </ns: MergeRecordsWS_MergeRecords_Output>

  </SOAP-ENV: Body>

</SOAP-ENV: Envelope>

```

MetadataChangeSummaryQueryPage

Gets a summary of changes to various categories of metadata.

Usage

You use the `MetadataChangeSummaryQueryPage` method to determine whether there have been any metadata changes in Oracle CRM On Demand. Only the relevant changes can be extracted and applied to client applications.

With the Administration Services API, field and picklist data, for example, can be accessed and modified (see, for example, [“FieldManagementUpsert” on page 240](#) and [“PicklistUpsert” on page 260](#)).

Arguments

[Table 54](#) describes the arguments taken by the `MetadataChangeSummaryQueryPage` method.

Table 54. Arguments Taken by the `MetadataChangeSummaryQueryPage` Method

Name	Description	Required	Default	I/O
ListOfMetadataChangeSummary	The metadata to be queried and returned.	Yes	Not applicable	Input/Output
recordcountneeded	Indicates whether a record count for the query is to be returned.	No	false	Input
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the method starts to return records.	No	0	Input

Table 54. Arguments Taken by the MetadataChangeSummaryQueryPage Method

Name	Description	Required	Default	I/O
searchspec	Indicates the search specification.	No	Not applicable	Input
sortsequence	An integer value that determines the order of the sort specification.	No		Input
sortorder	Determines the sort order for the records returned by the query, either ASC for ascending or DESC for descending.	No	ASC	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output
recordcount	An integer value that indicates the record count.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned:

- **LOVLastUpdated.** The date and time of the last picklist update.
- **CascPicklistsLastUpdated.** The date and time of the last cascading picklist update.
- **FieldManagementLastUpdated.** The date and time of the last field management data update.
- **WorkflowLastUpdated.** The date and time of the last workflow update.
- **AccessProfileLastUpdated.** The date and time of the last access profile update.
- **BookOfBusinessLastUpdated.** The date and time of the last book update.
- **ThemesLastUpdated.** The date and time of the last theme update.

Sample SOAP Request - MetadataChangesSummaryQueryPage

The following SOAP request retrieves the summary of changes to various categories of metadata:

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <body>

    <MetadataChangesSummaryQueryPage>

      <searchspec>

        <searchspec>

          <searchspec>

            <searchspec>

              <searchspec>

                <searchspec>

                  <searchspec>

                    <searchspec>

                      <searchspec>

                        <searchspec>

                          <searchspec>

                        </searchspec>

                      </searchspec>

                    </searchspec>

                  </searchspec>

                </searchspec>

              </searchspec>

            </searchspec>

          </searchspec>

        </searchspec>

      </searchspec>

      <sortsequence>

        <sortsequence>

          <sortsequence>

            <sortsequence>

              <sortsequence>

                <sortsequence>

                  <sortsequence>

                    <sortsequence>

                      <sortsequence>

                    </sortsequence>

                  </sortsequence>

                </sortsequence>

              </sortsequence>

            </sortsequence>

          </sortsequence>

        </sortsequence>

      </sortsequence>

      <sortorder>

        <sortorder>

          <sortorder>

            <sortorder>

              <sortorder>

                <sortorder>

                  <sortorder>

                    <sortorder>

                      <sortorder>

                    </sortorder>

                  </sortorder>

                </sortorder>

              </sortorder>

            </sortorder>

          </sortorder>

        </sortorder>

      </sortorder>

      <LastPage>

        <LastPage>

          <LastPage>

            <LastPage>

              <LastPage>

                <LastPage>

                  <LastPage>

                    <LastPage>

                      <LastPage>

                    </LastPage>

                  </LastPage>

                </LastPage>

              </LastPage>

            </LastPage>

          </LastPage>

        </LastPage>

      </LastPage>

      <recordcount>

        <recordcount>

          <recordcount>

            <recordcount>

              <recordcount>

                <recordcount>

                  <recordcount>

                    <recordcount>

                      <recordcount>

                    </recordcount>

                  </recordcount>

                </recordcount>

              </recordcount>

            </recordcount>

          </recordcount>

        </recordcount>

      </recordcount>

    </MetadataChangesSummaryQueryPage>

  </body>

</soap:Envelope>
```

```

    </wsse: UsernameToken>
  </wsse: Security>
</soap: Header>
<soap: Body>
  <MetadataChangeSummaryQueryPage_Input xmlns="urn: crmondemand/ws/metadatachangesummary/">
    <ListOfMetadataChangeSummary>
      <MetadataChangeSummary>
        <LOVLastUpdated />
        <CascadeListsLastUpdated />
        <FieldManagementLastUpdated />
        <WorkflowLastUpdated />
        <AccessProfileLastUpdated />
        <BookOfBusinessLastUpdated />
        <ThemesLastUpdated />
      </MetadataChangeSummary>
    </ListOfMetadataChangeSummary>
  </MetadataChangeSummaryQueryPage_Input>
</soap: Body>
</soap: Envelope>

```

Sample SOAP Response

The following response contains the returned metadata:

```

<?xml version="1.0" encoding="utf-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: MetadataChangeSummaryQueryPage_Output xmlns: ns="urn: crmondemand/ws/metadatachangesummary/">
      <ListOfMetadataChangeSummary xmlns="urn: /crmondemand/xml /metadatachangesummary/Data" lastpage="true">
        <MetadataChangeSummary>
          <LOVLastUpdated>2011-08-04T14: 46: 19</LOVLastUpdated>
          <CascadeListsLastUpdated>2011-08-04T15: 06: 08</CascadeListsLastUpdated>
          <FieldManagementLastUpdated>2011-08-03T11: 36: 21</FieldManagementLastUpdated>
          <WorkflowLastUpdated>2016-01-29T11: 32: 09</WorkflowLastUpdated>
          <AccessProfileLastUpdated>2016-12-01T20: 25: 14</AccessProfileLastUpdated>
          <BookOfBusinessLastUpdated>2016-12-19T11: 43: 06</BookOfBusinessLastUpdated>
        </MetadataChangeSummary>
      </ListOfMetadataChangeSummary>
    </ns: MetadataChangeSummaryQueryPage_Output>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

```

</ns:MetadataChangeSummaryQueryPage_Output>

</SOAP-ENV: Body>

</SOAP-ENV: Envelope>

```

SalesProcessQueryPage

Returns sales process information for a particular user, the default sales process, or all sales processes for a company.

Usage

You use the `SalesProcessQueryPage` method to retrieve sales process information including sales stages, order of the sales stages, default probability, and associated opportunity types.

The `ListOfSalesProcess` argument contains elements for the various fields of a sales process, and for the associated sales stages and opportunity types. There are child elements `ListofSalesStage` and `ListofOpportunityType`.

In requests, if the `<SalesProcessId>` element does not contain a value, information for all of the sales processes for the company are returned.

If the `<SalesProcessId>` element contains a value, information for sales processes for the user or role with that sales process ID are returned. You can use the User service to query for the `SalesProcId` value for a user.

If the `<Default>` element contains the value Y, information for the default sales process is returned.

For most of the child elements of `ListofSalesProcess`, you can use the `sortorder` and `sortsequence` attributes to specify how fields in the returned sales process information are sorted.

NOTE: The Name and Description fields of `SalesProcess` do not support the sort specification.

Arguments

[Table 55](#) describes the arguments taken by the `SalesProcessQueryPage` method.

Table 55. Arguments Taken by the `SalesProcessQueryPage` Method

Name	Description	Required	Default	I/O
ListofSalesProcess	The list of sales processes queried (input), and after query execution, the list of sales processes returned (output).	Yes	Not applicable	Input/Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input

Table 55. Arguments Taken by the SalesProcessQueryPage Method

Name	Description	Required	Default	I/O
StartRowNum	Indicates the row from which the SalesProcessQueryPage method starts to return records. For example, if you want to return records 1-100, you set StartRowNum to 0. Then, if you want to return records 101-200, you set StartRowNum to 100, and run the query again. You continue doing this until the last page is returned. In this way, you can return all records for a particular query.	No	0	Input
recordcountneeded	Indicates whether a record count for the object is to be returned.	No	False	Input
searchspec	Indicates the search specification for a field or for all the fields of an object.	No	Not applicable	Input
sortsequence	An integer value that determines the order of the sort specification.	No	Not applicable	Input
sortorder	Determines the sort order for the records returned by the query, either ASC for ascending or DESC for descending.	No	ASC	Input
LOVLanguageMode	The language mode for picklists, see “The LOVLanguageMode Argument” on page 104 .	No	LIC	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output
recordcount	An integer value that indicates the record count for the object	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned by ListOfSalesProcess which contains a <SalesProcess> element for each sales process that matches the query:

- **Description.** The description of the sales process.
- **Name.** The sales process name.
- **Translate.** Whether the sales process is marked for translation.
- **SalesProcessId.** The sales process ID.

- **Default.** Whether this sales process is the default sales process.
- **ListOfOpportunityType.** Information contained in the following child elements for each opportunity type:
 - **Id.** The opportunity type ID.
 - **Type.** The opportunity type name.
- **ListOfSalesStage.** Information contained in the following child elements for each sales stage:
 - **Name.** The name of the sales stage.
 - **SalesCategoryName.** The category that the sales stage falls under.
 - **Order.** The sequence of the sales stages as they appear in the picklist.
 - **Probability.** The default probability for the sales stage.
 - **Description.** The description of the sales stage.
 - **TranslationLanguage.** The translation language.

For more information about sales process information, see *Oracle CRM On Demand Online Help*.

Sample SOAP Requests and Responses

This topic contains sample SOAP requests that invoke SalesProcessQueryPage.

Sample SOAP Request 1 - SalesProcessQueryPage with pagesize, startrownum and recordcountneeded Attributes

The following SOAP request retrieves the sales processes defined for the company but only returns a maximum of one record at the SalesProcess parent object level, a maximum of one record at the OpportunityType child object level, and a maximum of five records at the SalesStage level by specifying pagesize, startrownum, and recordcountneeded attributes at each of the list of objects levels.

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

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

  <soap:Header>

    <wss:Security>

      <wss:UsernameToken>

        <wss:Username>%%USERNAMER%%</wss:Username>

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

      </wss:UsernameToken>

    </wss:Security>

  </soap:Header>

  <soap:Body>
```

```
<SalesProcessQueryPage_Input xmlns="urn:crmondemand/ws/salesproc/">
  <ListOfSalesProcess pagesize="1" startrownum="1" recordcountneeded="true">
    <SalesProcess>
      <Id />
      <Name />
      <Description />
      <Default />
      <ListOfOpportunityType pagesize="1" startrownum="0" recordcountneeded="false">
        <OpportunityType>
          <Id />
          <Type />
        </OpportunityType>
      </ListOfOpportunityType>
      <ListOfSalesStage pagesize="5" startrownum="0" recordcountneeded="true">
        <SalesStage>
          <Name />
        </SalesStage>
      </ListOfSalesStage>
    </SalesProcess>
  </ListOfSalesProcess>
</SalesProcessQueryPage_Input>
</soap:Body>
</soap:Envelope>
```

Sample SOAP Response 1

The following is the response for sample SOAP request 1:

```
<?xml version="1.0" encoding="utf-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:SalesProcessQueryPage_Output xmlns:ns="urn:crmondemand/ws/salesproc/">
      <ListOfSalesProcess xmlns="urn:/crmondemand/xml/salesprocess/Data" recordcount="2" lastpage="true">
        <SalesProcess>
          <Id>HRZ780-IDIT2</Id>
          <Name>Renewal Sales Process</Name>
          <Default>N</Default>
          <Description>Simplified process for renewal sales.</Description>
```

```

<ListOfOpportunityType lastpage="true">
  <OpportunityType>
    <Id>HRZ780-IDI V0</Id>
    <Type>Renewal </Type>
  </OpportunityType>
</ListOfOpportunityType>
<ListOfSalesStage recordcount="4" lastpage="true">
  <SalesStage>
    <Name>Re-Engage</Name>
  </SalesStage>
  <SalesStage>
    <Name>Proposal </Name>
  </SalesStage>
  <SalesStage>
    <Name>Closed/Won</Name>
  </SalesStage>
  <SalesStage>
    <Name>Closed/Lost</Name>
  </SalesStage>
</ListOfSalesStage>
</SalesProcess>
</ListOfSalesProcess>
</ns:SalesProcessQueryPage_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

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

The following SOAP request queries the sales process by specifying the search criteria in the searchspec attribute. It returns all sales processes with a name like Default* and all child sales stages of the parent sales process with a name like Closed*. It also demonstrates using the sortorder and sortsequence attributes to sort the result set.

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>

```

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

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

</wsse: UsernameToken>

</wsse: Security>

</soap: Header>

<soap: Body>

<SalesProcessQueryPage_Input xmlns="urn:crmondemand/ws/salesproc/">

<ListOfSalesProcess recordcountneeded="true">

<SalesProcess searchspec="[Name] LIKE 'Default*'">

  <Id />

  <Name />

  <Description />

  <Default />

  <ListOfOpportunityType>

    <OpportunityType>

      <Id />

      <Type />

    </OpportunityType>

  </ListOfOpportunityType>

  <ListOfSalesStage recordcountneeded="true">

    <SalesStage searchspec="[Name] LIKE 'Closed*'">

      <Name sortorder="ASC" sortsequence="0"></Name>

    </SalesStage>

  </ListOfSalesStage>

</SalesProcess>

</ListOfSalesProcess>

</SalesProcessQueryPage_Input>

</soap: Body>

</soap: Envelope>
```

Sample SOAP Response 2

The following is the response for sample SOAP request 2:

```
<?xml version="1.0" encoding="utf-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: SalesProcessQueryPage_Output xmlns:ns="urn: crmondemand/ws/salesproc/">
  <ListOfSalesProcess xmlns="urn: /crmondemand/xml /salesprocess/Data" recordcount="1" lastpage="true">
    <SalesProcess>
      <Id>HRZ780-I5RS0</Id>
      <Name>Default Sales Process</Name>
      <Default>Y</Default>
      <Description>This sales process is the default defined for your company.</Description>
      <ListOfOpportunityType lastpage="true">
        <OpportunityType>
          <Id>HRZ780-IDIVS</Id>
          <Type>New Business</Type>
        </OpportunityType>
      </ListOfOpportunityType>
      <ListOfSalesStage recordcount="2" lastpage="true">
        <SalesStage>
          <Name>Closed/Lost</Name>
        </SalesStage>
        <SalesStage>
          <Name>Closed/Won</Name>
        </SalesStage>
      </ListOfSalesStage>
    </SalesProcess>
  </ListOfSalesProcess>
</ns: SalesProcessQueryPage_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

SetPassword

Allows the system administrator to set the passwords of users who use Oracle CRM On Demand.

Usage

You use the SetPassword method to enable external applications to synchronize user passwords. For security reasons the password API is not available by default. If customers want to use SetPassword, they can call Customer Care to have the functionality enabled. There are two privileges:

- **Change Company Passwords using Web Services.** Enables access to SetPassword.

- **Manage Company Password Access.** Users (usually administrators) who have been granted this privilege can grant other roles the Change Company Passwords using Web Services privilege.

The API allows for the setting of passwords for one or more users at the same time. For each password that is updated, a corresponding user Audit Trail record is created. A user with the ability to set passwords does not have the ability to update the password of another user who has the ability to set passwords.

Arguments

Table 56 describes the arguments taken by SetPassword. These are child elements of <ListofUser>.

Allows the system administrator to set the passwords of users who use Oracle CRM On Demand.

Table 56. Arguments Taken by SetPassword

Field Name	Description	Required	Default	I/O
UserId	The row ID value of the user record.	Yes	Not applicable	Input/Output
EmailAddr	The user's email address	No	Not applicable	Input/Output
ExternalSystemId	The external system Id for the user.	No	Not applicable	Input/Output
IntegrationId	The integration Id	No	Not applicable	Input/Output
Password	The password for the user.	Yes	Not applicable	Input/Output

Sample SOAP Request - SetPassword

The following SOAP request updates the password for the users in the request.

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

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

  <soap:Header>

    <wss:Security>

      <wss:UsernameToken>

        <wss:Username>%%USERNAME%%</wss:Username>

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

      </wss:UsernameToken>

    </wss:Security>

  </soap:Header>
```

```

<soap: Body>
  <PasswordWS_SetPassword_Input xmlns="urn: crmondemand/ws/password/">
    <ListOfUser>
      <User>
        <UserId>HRZ780-KGY67</UserId>
        <EmailAddr />
        <Password>oracl etest123</Password>
        <IntegrationId />
        <ExternalSystemId />
      </User>
    </ListOfUser>
  </PasswordWS_SetPassword_Input>
</soap: Body>
</soap: Envelope>

```

Sample SOAP Response

The following response contains the status of the request:

```

<?xml version="1.0" encoding="utf-8"?>
<SOAP-ENV: Envelope xmlns: SOAP-ENV="http://schemas.xml soap.org/soap/envelope/" xmlns: xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns: xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV: Body>
    <ns: PasswordWS_SetPassword_Output xmlns: ns="urn: crmondemand/ws/password/">
      <ns: Status>Success</ns: Status>
    </ns: PasswordWS_SetPassword_Output>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

SetSessionTimeZone

Sets the time zone for a Web Services API session.

Usage

This method sets the time zone for a Web services API session. The time zone is set according to the locale of the user making the request.

Arguments

Table 57 describes the arguments taken by the SetSessionTimeZone method.

Table 57. Arguments Taken by the SetSessionTimeZone Method

Name	Description	Required	Default	I/O
TimeZone	The time zone of the user.	Yes	Not applicable	Input
CurrentServerTime	The server time zone.	Not applicable	Not applicable	Output

Return Value of the Call

The current server time zone.

Sample SOAP Request - SetSessionTimeZone

The following SOAP request updates the timezone for the current session to (GMT -06:00) Central Time (US & Canada).

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <TimeWS_SetSessionTimeZone_Input xmlns="urn:crmondemand/ws/time/">

      <TimeZone>(GMT-06:00) Central Time (US & Canada)</TimeZone>

    </TimeWS_SetSessionTimeZone_Input>

  </soap:Body>

</soap:Envelope>
```

Sample SOAP Response

The following response returns the server time zone set by the request:

```
<?xml version="1.0" encoding="utf-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:TimeWS_SetSessionTimeZone_Output xmlns:ns="urn:crmondemand/ws/time/">
      <ns:CurrentServerTime>(GMT-06:00) Central Time (US & Canada)</ns:CurrentServerTime>
    </ns:TimeWS_SetSessionTimeZone_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

UpdatePicklist

Updates picklist values in Oracle CRM On Demand.

Objects Supported

UpdatePicklist is supported for the same objects as supported by the GetPicklistValues method, see [“GetPicklistValues” on page 173](#).

Usage

You use the UpdatePicklist method to update lists of values, typically in a language-dependent manner.

The ListOfPicklistValues argument contains a sequence of PicklistValues elements, each of which has the following child elements:

- **DisplayValue.** The display value in the specified language.
- **Code.** The Language Independent Code (LIC).
- **Order.** The order of the value in the list.
- **NeedTranslate.** Whether the value is needed for translation.
- **Disabled.** Whether the value is disabled.

If the CreateNew argument is set to Y, new picklist values are created with the specified values.

If the CreateNew argument is set to Y, new picklist values are created with the specified values from DisplayValue and Order, which are required values.

If the CreateNew argument is set to N, existing picklist values are updated with the specified values from DisplayValue or Code, as long as these values are valid.

You cannot use the UpdatePicklist method to create custom picklists or multiselect picklists. You must create new picklist fields through the Oracle CRM On Demand application UI.

You cannot update read-only picklist fields.

Arguments

Table 58 describes the arguments taken by the UpdatePicklist method.

Table 58. 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	<i>User's Default Language</i>	Input
Status	A string indicating the success of the call or an error message if unsuccessful.	Not applicable	Not applicable	Output

Return Value of the Call

Returns a status string indicating the success or otherwise of the call.

Sample SOAP Requests and Responses

This topic contains sample SOAP requests that invoke UpdatePicklist.

Sample SOAP Request 1 - UpdatePicklist: Create a New Picklist Value

The following SOAP request creates a new picklist value for the CallFrequency picklist of the account record type.

NOTE: When you create new picklist values, the <Code> element is not taken into consideration, and the LIC will be equal to the <DisplayValue> value.

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

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%USERNAME%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <PicklistWS_UpdatePicklistInput xmlns="urn:crmondemand/ws/picklist/">
      <RecordType>Account</RecordType>
      <FieldName>CallFrequency</FieldName>
      <CreateNew>Y</CreateNew>
      <ListOfPicklistValues>
        <PicklistValues>
          <Code>BiYearly</Code>
          <DisplayValue>BiYearly_v</DisplayValue>
          <NeedTranslate>N</NeedTranslate>
          <Order>3</Order>
          <Disabled>N</Disabled>
        </PicklistValues>
      </ListOfPicklistValues>
    </PicklistWS_UpdatePicklistInput>
  </soap:Body>
</soap:Envelope>

```

Sample SOAP Request 2 - UpdatePicklist: Update an Existing Picklist Value

The following SOAP request updates an existing picklist value for the CallFrequency picklist of the account record type. The picklist value is updated in the user's default language:

```

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

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

```

```

    <wsse: UsernameToken>
      <wsse: Username>%%USERNAME%%</wsse: Username>
      <wsse: Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
        profile-1.0#PasswordText">%%PASSWORD%%</wsse: Password>
    </wsse: UsernameToken>
  </wsse: Security>
</soap: Header>
<soap: Body>
  <PicklistWS_UpdatePicklistInput xmlns="urn:crmondemand/ws/picklist/">
    <RecordType>Account</RecordType>
    <FieldName>CallFrequency</FieldName>
    <ListOfPicklistValues>
      <PicklistValues>
        <Code>BiYearly_v</Code>
        <DisplayValue>BiYearly</DisplayValue>
        <Order>3</Order>
        <Disabled>N</Disabled>
      </PicklistValues>
    </ListOfPicklistValues>
  </PicklistWS_UpdatePicklistInput>
</soap: Body>
</soap: Envelope>

```

Sample SOAP Request 3 - UpdatePicklist with LanguageCode

The following SOAP request updates an existing picklist value for the CallFrequency picklist of the account record type. The picklist value is updated in French according to the <LanguageCode> value specified in the 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:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
  secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap: Header>
    <wsse: Security>
      <wsse: UsernameToken>
        <wsse: Username>%%USERNAME%%</wsse: Username>
        <wsse: Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
          profile-1.0#PasswordText">%%PASSWORD%%</wsse: Password>
      </wsse: UsernameToken>
    </wsse: Security>

```



```

</soap: Header>
<soap: Body>
  <PicklistWS_UpdatePicklistInput xmlns="urn:crmondemand/ws/picklist/">
    <RecordType>Account</RecordType>
    <FieldName>CallFrequency</FieldName>
    <CreateNew>N</CreateNew>
    <LanguageCode>FRA</LanguageCode>
    <ListOfPicklistValues>
      <PicklistValues>
        <Code>BiYearly_v</Code>
        <DisplayValue>BiYearly_FRA</DisplayValue>
        <NeedTranslate>N</NeedTranslate>
        <Order>3</Order>
        <Disabled>N</Disabled>
      </PicklistValues>
    </ListOfPicklistValues>
  </PicklistWS_UpdatePicklistInput>
</soap: Body>
</soap: Envelope>

```

Sample SOAP Requests

The response is the same for each of the three sample SOAP requests:

```

<?xml version="1.0" encoding="utf-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:PicklistWS_UpdatePicklistOutput xmlns:ns="urn:crmondemand/ws/picklist/">
      <ns:Status>TRUE</ns:Status>
    </ns:PicklistWS_UpdatePicklistOutput>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

UpdateCascadingPicklists

Updates cascading picklist values in Oracle CRM On Demand.

Objects Supported

UpdateCascadingPicklists is supported for the same objects as supported by the GetPicklistValues method, see [“GetPicklistValues” on page 173](#).

Usage

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

The ListOfCascadingPicklistsValue argument contains a sequence of ParentCascPicklistsValue elements, which have the following child elements:

- **ParentDisplayValue.** A display value in the specified language.
- **ParentCode.** A parent Language Independent Code (LIC).
- **ListOfChildPickListValue.** A sequence of ChildPicklistValue elements containing the related picklist values that correspond to the parent picklist value.

The ChildPicklistValue element contains the following child elements:

- **DisplayValue.** The display value in the specified language.
- **Code.** The Language Independent Code (LIC).

If the CreateNew argument is set to Y, a new set of cascading picklist relationships is created with the specified values.

You cannot create a new Picklist field using the UpdateCascadingPicklist method.

Arguments

[Table 59](#) describes the arguments taken by the UpdateCascadingPicklists method.

Table 59. 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

Table 59. Arguments Taken by the UpdateCascadingPicklists Method

Name	Description	Required	Default	I/O
Description	A description of the cascading picklist.	No	Not applicable	Input
ListOfCascadingPicklistValue	A sequence of ParentCascPicklistValue elements containing the parent picklist values to be updated or added.	Yes	Not applicable	Input
LanguageCode	The language code for the picklist values to be updated, for example, ENU, DEU, FRA, ESN, and so on. If the code is not specified, the default language for the current session's user is used.	No	<i>User's Default Language</i>	Input
Status	A string indicating the success of the call or an error message if unsuccessful.	Not applicable	Not applicable	Output

Return Value of the Call

Returns a status string indicating the success or otherwise of the call.

Sample SOAP Requests and Responses

This topic contains sample SOAP requests that invoke UpdateCascadingPicklists.

Sample SOAP Request 1 - UpdateCascadingPicklist: Create a New Cascading Picklist

The following SOAP request creates a new cascading picklist for the Account record type:

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <PicklistWS_UpdateCascadingPicklists_Input xmlns="urn:crmondemand/ws/picklist/">
```

```

<RecordType>Account</RecordType>
<ParentFiel dName>AccountType</ParentFiel dName>
<Fiel dName>Pri ori ty</Fiel dName>
<CreateNew>Y</CreateNew>
<Li stOfCascadi ngPi ckl i stsVal ue>
  <ParentCascPi ckl i stsVal ue>
    <ParentCode>Competi tor</ParentCode>
    <ParentDi spl ayVal ue>Competi tor</ParentDi spl ayVal ue>
    <Li stOfChi l dPi ckl i stVal ue>
      <Chi l dPi ckl i stVal ue>
        <Code>Low</Code>
      </Chi l dPi ckl i stVal ue>
    </Li stOfChi l dPi ckl i stVal ue>
  </ParentCascPi ckl i stsVal ue>
  <ParentCascPi ckl i stsVal ue>
    <ParentCode>Customer</ParentCode>
    <Li stOfChi l dPi ckl i stVal ue>
      <Chi l dPi ckl i stVal ue>
        <Code>Low</Code>
        <Di spl ayVal ue>Low</Di spl ayVal ue>
      </Chi l dPi ckl i stVal ue>
      <Chi l dPi ckl i stVal ue>
        <Code>Hi gh</Code>
        <Di spl ayVal ue>Hi gh</Di spl ayVal ue>
      </Chi l dPi ckl i stVal ue>
    </Li stOfChi l dPi ckl i stVal ue>
  </ParentCascPi ckl i stsVal ue>
</Li stOfCascadi ngPi ckl i stsVal ue>
</Pi ckl i stWS_UpdateCascadi ngPi ckl i sts_I nput>
</soap: Body>
</soap: Envel ope>

```

Sample SOAP Request 2 - UpdateCascadingPicklist: Update an Existing Cascading Picklist

The following SOAP request updates an existing cascading picklist for the account record type. The picklist value is updated in the user's default language:

```
<?xml versi on="1.0" encodi ng="utf-8" standal one="no"?>
```

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

```
<soap:Header>
```

```
<wsse:Security>
```

```
<wsse:UsernameToken>
```

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

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

```
</wsse:UsernameToken>
```

```
</wsse:Security>
```

```
</soap:Header>
```

```
<soap:Body>
```

```
<PicklistWS_UpdateCascadingPicklistInput xmlns="urn:crmondemand/ws/picklist/">
```

```
<RecordType>Account</RecordType>
```

```
<ParentFieldName>AccountType</ParentFieldName>
```

```
<FieldName>Priority</FieldName>
```

```
<ListOfCascadingPicklistValues>
```

```
<ParentCascPicklistValues>
```

```
<ParentCode>Partner</ParentCode>
```

```
<ParentDisplayValue>Partner</ParentDisplayValue>
```

```
<ListOfChildPicklistValues>
```

```
<ChildPicklistValues>
```

```
<Code>Medium</Code>
```

```
</ChildPicklistValues>
```

```
</ListOfChildPicklistValues>
```

```
</ParentCascPicklistValues>
```

```
<ParentCascPicklistValues>
```

```
<ParentCode>Customer</ParentCode>
```

```
<ListOfChildPicklistValues>
```

```
<ChildPicklistValues>
```

```
<Code>High</Code>
```

```
<DisplayValue>High</DisplayValue>
```

```
</ChildPicklistValues>
```

```
</ListOfChildPicklistValues>
```

```
</ParentCascPicklistValues>
```

```
</ListOfCascadingPicklistValues>
```

```
</PicklistWS_UpdateCascadingPicklistInput>
```

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

Sample SOAP Responses

The response is the same for both of the sample SOAP requests:

```
<?xml version="1.0" encoding="utf-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: PickListWS_UpdateCascadingPickLists_Output xmlns:ns="urn:crmondemand/ws/picklist/">
      <ns: Status>TRUE</ns: Status>
    </ns: PickListWS_UpdateCascadingPickLists_Output>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

UserUsageQueryPage

Executes a query against the list of Web Services utilization, and returns a subset of the records that match the search criteria set by the method arguments.

Usage

You use the `UserUsageQueryPage` method to query the details of your company's Web services utilization. This is the same data that you can view on the Web Services Utilization page in the Oracle CRM On Demand application. As an example of how you might use this data, you might save the data in a CSV file and then import it into a spreadsheet. You could then use the spreadsheet to generate a report showing, for example, how often a user updates his or her records.

In the input request, if you supply a value for `UserId` in the `ListOfUserUsage` argument, the method returns the utilization records for the specified user. If you do not have Administrator privileges, you can only specify your own user ID.

If you do not supply a value for `UserId` in the input request, the method returns:

- The utilization records for the current user, if you do not have Administrator privileges.
- The utilization records for the whole company, if you do have Administrator privileges.

Arguments

Table 60 describes the arguments taken by the `UserUsageQueryPage` method.

Table 60. Arguments Taken by the `UserUsageQueryPage` Method

Name	Description	Required	Default	I/O
ListOfUserUsage	The list of Web service utilization queried (input), and after query execution, the list of Web service utilization returned (output).	Yes	Not applicable	Input/Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the method starts to return records.	No	0	Input
LastPage	A value that indicates whether or not the last value in the query set has been returned.	Not applicable	Not applicable	Output

Return Value of the Call

The following information is returned for each usage record.

A list of Web service utilization. The following are the child elements of `ListOfUserUsage`:

- **SessionId**. The session identifier of the Web service request.
- **UserAlias**. The user alias of the user who executed the Web service request.
- **UserId**. The user ID of the user who executed the Web service request.
- **WebServiceName**. The name of the Web service to which the request was made.
- **WebServiceNameSpace**. The namespace used in the request.
- **Operation**. The operation for the Web service request.
- **StartTime**. The start time of the Web service request.
- **EndTime**. The end time of the Web service request.
- **EntryType**. The entry type for the Web service request.
- **InputMessageSize**. The size of the input message.
- **OutputMessageSize**. The size of the output message.
- **ErrorMsg**. Any error message associated with the Web service request.

Sample SOAP Request - UserUsageQueryPage

The following SOAP request returns the list of Web services utilization records. It also demonstrates the use of the optional <PageSize> and <StartRowNum> arguments to return only five records starting at row 2 (third record).

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

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

  <soap:Header>

    <wsse:Security>

      <wsse:UsernameToken>

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

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

      </wsse:UsernameToken>

    </wsse:Security>

  </soap:Header>

  <soap:Body>

    <UserUsageWS_UserUsageQueryPage_Input xmlns="urn:crmondemand/ws/userusage/">

      <PageSize>5</PageSize>

      <StartRowNum>2</StartRowNum>

      <ListOfUserUsage>

        <UserUsage>

          <SessionId />

          <UserAlias />

          <UserId />

          <WebServiceName />

          <WebServiceNameSpace />

          <Operation />

          <StartTime />

          <EndTime />

          <EntryType />

          <InputMessageSize />

          <OutputMessageSize />

          <ErrorMsg />

        </UserUsage>

      </ListOfUserUsage>

    </UserUsageWS_UserUsageQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```



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

Sample SOAP Response

The following illustrates the SOAP response. For brevity, a number of <UserUsage> elements are omitted as indicated by the ellipses (...).

```
<?xml version="1.0" encoding="utf-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: UserUsageWS_UserUsageQueryPage_Output xmlns:ns="urn: crmondemand/ws/userusage/">
      <ns: LastPage>false</ns: LastPage>
      <ListOfUserUsage xmlns="urn: /crmondemand/xml /userusage">
        <UserUsage>
          <SessionId>ADSA-920RIQ</SessionId>
          <UserAlias>Joanne Brown</UserAlias>
          <UserId>HRZ780-I5RUE</UserId>
          <WebServiceName></WebServiceName>
          <WebServiceNameSpace></WebServiceNameSpace>
          <Operation>Login</Operation>
          <StartTime>08/02/2011 13: 53: 00</StartTime>
          <EndTime>08/02/2011 13: 53: 00</EndTime>
          <EntryType>Login</EntryType>
          <InputMessageSize></InputMessageSize>
          <OutputMessageSize></OutputMessageSize>
          <ErrorMsg></ErrorMsg>
        </UserUsage>
        ...
        ...
        <UserUsage>
          <SessionId>ADSA-921BAY</SessionId>
          <UserAlias>Joanne Brown</UserAlias>
          <UserId>HRZ780-I5RUE</UserId>
          <WebServiceName></WebServiceName>
          <WebServiceNameSpace></WebServiceNameSpace>
          <Operation>Logout</Operation>
          <StartTime>08/02/2011 14: 38: 15</StartTime>
```

```
<EndTime>08/02/2011 14:38:15</EndTime>
<EntryType>Logout</EntryType>
<InputMessageSize></InputMessageSize>
<OutputMessageSize></OutputMessageSize>
<ErrorMsg></ErrorMsg>
</UserUsage>
</ListOfUserUsage>
</ns:UserUsageWS_UserUsageQueryPage_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Administrative Services API Calls

The Administrative Services are used to automate the administration of your company's configurations. The Oracle Migration Tool On Demand client is the command-line based utility that eliminates the need to manually copy customized configurations from one environment to another. You can use this downloadable client to extract and import specific configuration details and the Administrative Web services provide programmatic access to the configurations.

NOTE: The Administrative Services must be enabled for your company. To request enablement of the Administrative Services, contact Customer Care.

For more information about the downloadable client, see *Oracle Migration Tool On Demand Guide*.

The Administrative Services and their methods are as shown in [Table 61](#). There are the following methods for each service, though not all services have each method, as shown in the table:

- **<Configuration>Read**. Extracts configuration data matching the input criteria.
- **<Configuration>ReadAll**. Extracts all configuration data of a particular type for a company.
- **<Configuration>Upsert**. Updates existing configuration data or inserts it, if it does not exist.
- **<Configuration>Create**. Creates configuration data.
- **<Configuration>Delete**. Deletes configuration data.

For access to configuration data, various privileges are required, depending on the type of data. The privileges are detailed in the *Oracle Migration Tool On Demand Guide*. Users with an Administrator role will have the necessary privileges to use all of the Administrative Services, although read-only access to some configuration data is available for users who are not administrators, see [“Access to the Administrative Services APIs for Nonadministrative Users” on page 213](#).

The Administrative Services do not support stateful login mechanisms, that is, Single Sign-on (SSO), and use of session IDs to authenticate users. For more information about stateful versus stateless session management, see [Chapter 4, “Establishing and Managing the Web Services Session.”](#)

Table 61. Administrative Services

Service	Method	Usage
Access Profile	AccessProfileRead	Administering access profile configuration data.
	AccessProfileReadAll	
	AccessProfileUpsert	
Action Bar Custom Web Applet	ActionBarCustomWebAppletRead	Administering configuration data for action bar Web applets.
	ActionBarCustomWebAppletReadAll	
	ActionBarCustomWebAppletUpsert	
Action Bar Layout	ActionBarLayoutReadAll	Administering action bar layout configuration data.
	ActionBarLayoutUpsert	
Assessment Scripts	SalesAssessmentTemplateRead	Administering assessment script configuration data.
	SalesAssessmentTemplateReadAll	
	SalesAssessmentTemplateUpsert	
Cascading Picklists	CascadingPicklistRead	Administering cascading picklist configuration data.
	CascadingPicklistReadAll	
	CascadingPicklistUpsert	
Company Profile	CurrentOrganizationRead	Administering the company profile data.
Currency Picklist	CurrencyRead	Administering currency picklist configuration data.
	CurrencyReadAll	
	CurrencyUpsert	
Custom Record Type	CustomRecordTypeRead	Administering custom record type configuration data.
	CustomRecordTypeReadAll	
	CustomRecordTypeUpsert	
Custom Web Link	CustomWebLinkRead	Administering Web link configuration data.
	CustomWebLinkReadAll	
	CustomWebLinkUpsert	
Custom Web Tab	CustomWebTabDelete	Administering custom Web tab configuration data.
	CustomWebTabRead	
	CustomWebTabReadAll	
	CustomWebTabUpsert	

Service	Method	Usage
Detail Page Custom Web Applet	DetailPageCustomWebAppletRead	Administering detail page Web applet configuration data
	DetailPageCustomWebAppletReadAll	
	DetailPageCustomWebAppletUpsert	
Field Management	FieldManagementCreate	Administering field management configuration data.
	FieldManagementRead	
	FieldManagementReadAll	
	FieldManagementUpsert	
Homepage Layout	HomepageLayoutReadAll	Administering homepage layout configuration data.
	HomepageLayoutUpsert	
Home Page Custom Web Applet	HomepageCustomWebAppletRead	Administering homepage Web applet configuration data.
	HomepageCustomWebAppletReadAll	
	HomepageCustomWebAppletUpsert	
Home Tab Custom Web Applet	HomeTabCustomWebAppletRead	Administering My Homepage tab Web applet configuration data.
	HomeTabCustomWebAppletReadAll	
	HomeTabCustomWebAppletUpsert	
Industry	IndustryRead	Administering industry configuration data.
	IndustryReadAll	
Page Layout Field	PageLayoutFieldRead	Administering field information for page layout configuration data.
	PageLayoutFieldReadAll	
	PageLayoutFieldUpsert	
Page Layout Related Information	PageLayoutRelatedInformationRead	Administering related information for page layout configuration data.
	PageLayoutRelatedInformationReadAll	
	PageLayoutRelatedInformationUpsert	
Page Layout Section	PageLayoutSectionReadAll	Administering section information for page layout configuration data.
	PageLayoutSectionUpsert	
Picklist	PicklistRead	Administering picklist configuration data.
	PicklistReadAll	
	PicklistUpsert	
Related Information Layout	RelatedInformationLayoutRead	Administering related information for page layout configuration data.
	RelatedInformationLayoutReadAll	
	RelatedInformationLayoutUpsert	

Service	Method	Usage
Role Management	RoleRead	Administering role management configuration data.
	RoleReadAll	
	RoleUpsert	
Workflow Action	WorkflowActionSetRead	Administering workflow action configuration data.
	WorkflowActionSetReadForRuleSet	
	WorkflowActionSetUpsert	
Workflow Rule	WorkflowRuleSetDelete	Administering workflow rule configuration data
	WorkflowRuleSetRead	
	WorkflowRuleSetReadAll	
	WorkflowRuleSetUpsert	

You can download the WSDL file for each service from the Web Services Administration page in the Oracle CRM On Demand application.

All Web Service operations for the Administrative Services are audited automatically. The audits include all successful operations as well as processing errors such as Request Size exceeded, Invalid SOAP request, Rate limit error, and so on. You can find the audit records in the Web Services Utilization page in Oracle CRM On Demand (click Admin, and then Web Services Utilization).

You can view logs for the Oracle Migration Tool On Demand client in the *<Installation directory>\log* directory, for example, C:\Oracle Migration Tool On Demand\log. Any requests by the client are also captured in the Web Services Utilization page in Oracle CRM On Demand.

The following topics describe each of the methods.

Access to the Administrative Services APIs for Nonadministrative Users

Users with an Administrator role have the necessary privileges to use all of the methods of the Administrative Services. However, for the following services, read-only access is available for users who do not have the Administrator role:

- Access Profile
- Action Bar Custom Web Applet
- Assessment Scripts
- Currency Picklist
- Custom Record Type
- Detail Page Custom Web Applet
- Field Management

- Home Page Custom Web Applet
- Home Tab Custom Web Applet
- Page Layout Field
- Page Layout Related Information
- Picklist
- Role Management
- Workflow Action
- Workflow Rule

Administrators can read, create, update and delete configurations. Other users can only issue Read and ReadAll requests for the services listed; they must only have been granted access to Web services to issue requests (the Enable Web Services Access privilege).

AccessProfileRead

Extracts access profile configuration data.

Usage

Use the AccessProfileRead method to extract access profile configuration data matching the input criteria.

The access profile is specified with the <Name> child element of apQuery:AccessProfile.

See [“AccessProfileUpsert” on page 215](#) for information about the access profile data specified in apData:AccessProfile.

Arguments

[Table 62](#) describes the arguments taken by the AccessProfileRead method.

Table 62. Arguments Taken by the AccessProfileRead Method

Name	Description	Required	Default	I/O
apQuery:AccessProfile	The access profile query criteria.	Yes	Not applicable	Input
apData:AccessProfile	The extracted access profile data.	Not applicable	Not applicable	Output

Return Value of the Call

The access profile is returned.

AccessProfileReadAll

Extracts all of the access profile configuration data for a company.

Usage

Use the AccessProfileReadAll method to read all of the access profiles.

There is no input argument; the method simply returns all of the access profile data.

See [“AccessProfileUpsert” on page 215](#) for information about the access profile data specified in apData:AccessProfile.

Arguments

[Table 63](#) describes the arguments taken by the AccessProfileReadAll method.

Table 63. Arguments Taken by the AccessProfileReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfAccessProfile	The extracted data for all access profiles.	Not applicable	Not applicable	Output

Return Value of the Call

The access profiles are returned.

AccessProfileUpsert

Updates an existing access profile or inserts a new access profile.

Usage

Use the AccessProfileUpsert method to insert a profile or update an existing access profile.

The apdata:AccessProfile argument contains the following elements containing access profile data:

- **Name.** The name of the access profile. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <AccessProfileTranslation> element.
- **Description.** A description of the access profile.
- **AvailableForTeam.** Whether the profile can be assigned to team members.
- **AvailableForBook.** Whether the profile can be assigned to book users.
- **Disabled.** Whether the profile cannot be assigned, but existing assignments continue to work.
- **ListOfAccessProfileTranslation.** The text fields in the languages activated for your company:
 - **LanguageCode.** The three-letter language code for the language.

- **Title.** The display name of the access profile in the language identified by the language code.
- **Description.** The description of the access profile in the language identified by the language code.
- **ListOfAccessProfileEntry.** The access rights defined on the access profile for each object and child object:
 - **AccessObjectName.** The names of objects as used in access profiles.
 - **PermissionCode.** The permission codes corresponding to the various access levels for objects.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of access profile object names to the display names for the objects in the Oracle CRM On Demand user interface.
- The permission codes for the various access levels in Oracle CRM On Demand.

Arguments

[Table 64](#) describes the arguments taken by the `AccessProfileUpsert` method.

Table 64. Arguments Taken by the `AccessProfileUpsert` Method

Name	Description	Required	Default	I/O
apData:AccessProfile	The access profile data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

ActionBarCustomWebAppletRead

Extracts action bar Web applet configuration data.

Usage

Use the `ActionBarCustomWebAppletRead` method to read configuration data for an action bar Web applet.

The Web applet is specified with the `<DisplayName>` child element of `apQuery:ActionBarCustomWebApplet`.

See [“ActionBarCustomWebAppletUpsert” on page 218](#) for information about the data specified in `apData: ListOfActionBarCustomWebApplet`.

Arguments

[Table 65](#) describes the arguments taken by the `ActionBarCustomWebAppletRead` method.

Table 65. Arguments Taken by the `ActionBarCustomWebAppletRead` Method

Name	Description	Required	Default	I/O
<code>apQuery: ActionBarCustomWebApplet</code>	The Web applet query criteria.	Yes	Not applicable	Input
<code>apData: ListOfActionBarCustomWebApplet</code>	The extracted data for the Web applet.	Not applicable	Not applicable	Output

Return Value of the Call

The Web applet data is returned.

ActionBarCustomWebAppletReadAll

Extracts all of the action bar Web applet configuration data for a company.

Usage

Use the `ActionBarCustomWebAppletReadAll` method to read all of the action bar Web applets.

There is no input argument; the method simply returns all of the Web applet data.

See [“ActionBarCustomWebAppletUpsert” on page 218](#) for information about the data specified in `apData: ListOfActionBarCustomWebApplet`.

Arguments

[Table 66](#) describes the arguments taken by the `ActionBarCustomWebAppletReadAll` method.

Table 66. Arguments Taken by the `ActionBarCustomWebAppletReadAll` Method

Name	Description	Required	Default	I/O
<code>apData: ListOfActionBarCustomWebApplet</code>	The extracted data for all action bar Web applets.	Not applicable	Not applicable	Output

Return Value of the Call

The action bar Web applets are returned.

ActionBarCustomWebAppletUpsert

Updates an existing action bar Web applet or inserts a new Web applet.

Usage

Use the ActionBarCustomWebAppletUpsert method to insert or update an existing action bar Web applet.

The apData:ActionBarCustomWebApplet argument contains the following elements containing action bar Web applet data:

- **DisplayName.** The name of the Web applet. Required.
- **Description.** A description for the Web applet.
- **Height.** The height of the applet in pixels.
- **FeedType.** Settings for an RSS feed applet:
 - **URL.** The URL that is invoked when the user clicks the hyperlink.
 - **OverrideURLFlag.** Whether or not the user can override the URL.
 - **TargetWindow.** The window in which the Web applet opens: Current Window or New Window.
- **HTMLType.** Settings for an HTML Web applet:
 - **HTMLHeader.** HTML code used in an <iframe> element within the HTML header of the page to which you add the Web applet.
 - **HTMLBody.** HTML code used in an <iframe> element within the HTML body of the page to which you add the Web applet.
 - **AlwaysRun.** Whether the Web applet will be executed if the applet is minimized.
- **URLType.** Settings for a URL Web applet:
 - **URL.** The URL that is invoked when the user clicks the hyperlink.
 - **AlwaysRun.** Whether the Web applet will be executed if the applet is minimized.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets.

Arguments

Table 67 describes the arguments taken by the ActionBarCustomWebAppletUpsert method.

Table 67. Arguments Taken by the ActionBarCustomWebAppletUpsert Method

Name	Description	Required	Default	I/O
apData:ActionBarCustomWebApplet	The Web applet data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

ActionBarLayoutReadAll

Extracts all of the action bar layout configuration data for a company.

Usage

Use the ActionBarLayoutReadAll method to read all of the action bar layouts.

There is no input argument; the method simply returns all of the action bar layout data.

See “[ActionBarLayoutUpsert](#)” on page 220 for information about the action bar layout data specified in ablData:ActionBarLayout.

Arguments

Table 68 describes the arguments taken by the ActionBarLayoutReadAll method.

Table 68. Arguments Taken by the ActionBarLayoutReadAll Method

Name	Description	Required	Default	I/O
ablData:ListOfActionBarLayout	The extracted data for all action bar layouts.	Not applicable	Not applicable	Output

Return Value of the Call

The action bar layouts are returned.

ActionBarLayoutUpsert

Updates an existing action bar layout or inserts a new action bar layout.

Usage

Use the `ActionBarLayoutUpsert` method to insert or update an existing action bar layout.

The `abldata:ActionBarLayout` argument contains the following elements containing action bar layout data:

- **LayoutName.** The name of the action bar layout.
- **Description.** A description of the action bar layout.
- **ListOfSections.** Contains all of the available sections for the action bar layout. The data for a particular section is contained in the following elements:
 - **Name.** The name of the section, which can be one of the following:
 - ❑ Calendar Action Applet
 - ❑ Global Search Multi Field
 - ❑ Quick Create
 - ❑ Quick Favorite Lists
 - ❑ Quick Favorite Records
 - ❑ Quick History
 - ❑ Quick Message Center
 - **Displayed.** Whether the section is displayed in the action bar.
 - **Sequence.** The position of the section within the action bar layout. This corresponds to step 2 in the Action Bar Layout wizard in the UI. The integer value can be one of the following:
 - ❑ **-2.** The section is in the Not Available Sections list.
 - ❑ **-1.** The section is in the Available Sections list.
 - ❑ **Positive integers.** The section is in the Displayed Sections list. The integer value indicates the location of the section within the lists, with 0 (zero) meaning the section is located at the top of the list.

See *Oracle Migration Tool On Demand Guide* for information about mappings of action bar section names to the display names for the sections in the Oracle CRM On Demand user interface.

Arguments

Table 69 describes the arguments taken by the ActionBarLayoutUpsert method.

Table 69. Arguments Taken by the ActionBarLayoutUpsert Method

Name	Description	Required	Default	I/O
ablData:ActionBarLayout	The action bar layout data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CascadingPicklistRead

Extracts cascading picklist configuration data for an object.

Usage

Use the CascadingPicklistRead method to extract the set of cascading picklist data matching the input criteria.

The object is specified with the <ObjectName> child element of apQuery:CascadingPicklistSet. You can also specify a particular parent picklist or related picklist in the query.

The apData:ListOfCascadingPicklistSet output argument can contain any number of CascadingPicklist elements, each of which contains the following data:

- **ParentPicklist.** The parent picklist.
- **RelatedPicklist.** The related picklist.
- **Description.** The description of the cascading picklist.
- **ListOfPicklistValueAssociations.** Contains all the associations for a cascading picklist. The associations for a particular picklist are contained in the following elements:
 - **ParentPicklistValue.** A parent picklist value for the cascading picklist.
 - **RelatedPicklistValue.** The related values for the parent picklist. There is one these elements for each related picklist value.

Arguments

[Table 70](#) describes the arguments taken by the CascadingPicklistRead method.

Table 70. Arguments Taken by the CascadingPicklistRead Method

Name	Description	Required	Default	I/O
apQuery:CascadingPicklistSet	The cascading picklist query criteria.	Yes	Not applicable	Input
apData:ListOfCascadingPicklistSet	The extracted cascading picklist data.	Not applicable	Not applicable	Output

Return Value of the Call

The cascading picklist data for the object is returned.

CascadingPicklistReadAll

Extracts all of the cascading picklist configuration data for a company.

Usage

Use the CascadingPicklistReadAll method to read all of the cascading picklists.

There is no input argument; the method simply returns all of the cascading picklist data for each object.

See [“CascadingPicklistRead” on page 221](#) for information about the cascading picklist data returned in apData:ListOfCascadingPicklistSet.

Arguments

[Table 71](#) describes the arguments taken by the CascadingPicklistReadAll method.

Table 71. Arguments Taken by the CascadingPicklistReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfCascadingPicklistSet	The extracted data for all cascading picklists.	Not applicable	Not applicable	Output

Return Value of the Call

The cascading picklists are returned.

CascadingPicklistUpsert

Updates cascading picklist configuration data for a company.

Usage

Use the CascadingPicklistUpsert method to insert or update a cascading picklist.

The apData:CascadingPicklistSet argument contains the following elements specifying the data for the cascading picklist:

- **ObjectName.** The name of the object.
- **ListOfCascadingPicklistSet.** Contains any number of CascadingPicklist elements, each of which contains the following data:
 - **ParentPicklist.** The parent picklist.
 - **RelatedPicklist.** The related picklist.
 - **Description.** The description of the cascading picklist.
 - **ListOfPicklistValueAssociations.** Contains all the associations for the cascading picklist. An associations is contained in the following elements:
 - **ParentPicklistValue.** A parent picklist value for the cascading picklist.
 - **RelatedPicklistValue.** The related values for the parent picklist. There is one these elements for each related picklist value.

Update occurs when a cascading picklist already exists with the specified <ParentPicklist> and <RelatedPicklist> values otherwise an Insert occurs.

Upsert fails and an error message is returned if:

- The specified parent picklist and related picklist are the same.
- The parent or related picklist references a field that does not exist.
- The parent or related picklist references a field that is not of type picklist.
- The related picklist references a parent picklist that is already referenced by another related picklist.

Arguments

[Table 72](#) describes the arguments taken by the CascadingPicklistUpsert method.

Table 72. Arguments Taken by the CascadingPicklistUpsert Method

Name	Description	Required	Default	I/O
apData:CascadingPicklistSet	The cascading picklist data to be updated or inserted.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The cascading picklists are returned.

CurrencyRead

Extracts currency configuration data.

Usage

Use the CurrencyRead method to extract currency configuration data matching the input criteria.

The currency is specified with the <Name> child element of apQuery:Currency.

See [“CurrencyUpsert” on page 225](#) for information about the currency data specified in apData:ListOfCurrency.

Arguments

[Table 73](#) describes the arguments taken by the CurrencyRead method.

Table 73. Arguments Taken by the CurrencyRead Method

Name	Description	Required	Default	I/O
apQuery:Currency	The currency query criteria.	Yes	Not applicable	Input
apData:ListOfCurrency	The extracted currency data.	Not applicable	Not applicable	Output

Return Value of the Call

The currency data is returned.

CurrencyReadAll

Extracts all of the currency configuration data for a company.

Usage

Use the CurrencyReadAll method to read all of the currency data.

There is no input argument; the method simply returns all of the currency data.

See [“CurrencyUpsert” on page 225](#) for information about the currency data specified in apData:ListOfCurrency.

Arguments

[Table 74](#) describes the arguments taken by the CurrencyReadAll method.

Table 74. Arguments Taken by the CurrencyReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfCurrency	The extracted data for all currencies.	Not applicable	Not applicable	Output

Return Value of the Call

The currency data is returned.

CurrencyUpsert

Updates existing currency data.

Usage

Use the CurrencyUpsert method to update currency data.

It is not possible to define a new currency or delete an existing currency in Oracle CRM On Demand, therefore CurrencyUpsert only supports update and not insertion of data. The only updating supported is the active/inactive setting and symbol for active currencies.

The apdata:Currency argument contains the following elements containing currency data:

- **Name.** The name of the currency.
- **Code.** The three-letter code for the currency.
- **Symbol.** The symbol for the currency.
- **IssuingCountry.** The country that issues the currency.

■ **Active.** Whether the currency is active for the company.

Arguments

Table 75 describes the arguments taken by the CurrencyUpsert method.

Table 75. Arguments Taken by the CurrencyUpsert Method

Name	Description	Required	Default	I/O
apData:Currency	The currency data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CurrentOrganizationRead

Extracts company profile data.

Usage

Use the CurrentOrganizationRead method to extract company profile data.

Arguments

Table 76 describes the arguments taken by the CurrentOrganizationRead method.

Table 76. Arguments Taken by the CurrentOrganizationRead Method

Name	Description	Required	Default	I/O
apData:ListOfOrganization	The extracted company profile data.	Not applicable	Not applicable	Output

Return Value of the Call

The company profile data is returned. See *Oracle CRM On Demand Online Help* for information about the company profile settings.

CustomRecordTypeRead

Extracts custom record type configuration data.

Usage

Use the CustomRecordTypeRead method to extract custom record type configuration data matching the input criteria.

The custom record type is specified with the <Name> child element of apQuery:CustomRecordType.

If the <IncludeAll> child element of apQuery:CustomRecordType specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See [“CustomRecordTypeUpsert” on page 228](#) for information about the custom record type data specified in apData:ListOfCustomRecordType.

Arguments

[Table 77](#) describes the arguments taken by the CustomRecordTypeRead method.

Table 77. Arguments Taken by the CustomRecordTypeRead Method

Name	Description	Required	Default	I/O
apQuery:CustomRecordType	The custom record type query criteria.	Yes	Not applicable	Input
apData:ListOfCustomRecordType	The extracted custom record type data.	Not applicable	Not applicable	Output

Return Value of the Call

The custom record type data is returned.

CustomRecordTypeReadAll

Extracts all of the custom record type configuration data for a company.

Usage

Use the CustomRecordTypeReadAll method to read all of the custom record types.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See [“CustomRecordTypeUpsert” on page 228](#) for information about the custom record type data specified in `apData:ListOfCustomRecordType`.

Arguments

[Table 78](#) describes the arguments taken by the `CustomRecordTypeReadAll` method.

Table 78. Arguments Taken by the CustomRecordTypeReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfCustomRecordType	The extracted data for all custom record types.	Not applicable	Not applicable	Output

Return Value of the Call

The custom record types are returned.

CustomRecordTypeUpsert

Updates an existing custom record type or inserts a new custom record type.

Usage

Use the `CustomRecordTypeUpsert` method to insert or update custom record type data.

The `apdata:CustomRecordType` argument contains the following elements containing custom record type data:

- **Name.** The object name of the custom record type.
 - **SingularName.** The singular display name used in the UI.
 - **PluralName.** The plural display name used in the UI.
 - **ShortName.** The short display name used in the UI.
- NOTE:** `SingularName`, `PluralName` and `ShortName` are language independent names. If these names are present in the input, the behavior is similar to Mark for Translation being selected in the UI. These elements are not required for customizing language dependent singular/plural/short names for an object, as they are handled by `ListOfCustomRecordTypeTranslation`.
- **IconName.** The filename of the icon for the custom object.
 - **ListOfCustomRecordTypeTranslation.** The object display names in the languages activated for your company:

- **LanguageCode.** The three-letter language code for the language.
- **SingularName.** The singular name of the custom record type in the language identified by the language code.
- **PluralName.** The plural name of the custom record type in the language identified by the language code.
- **ShortName.** The short name of the custom record type in the language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of icon filenames to icons for custom objects in the Oracle CRM On Demand user interface.

Arguments

[Table 79](#) describes the arguments taken by the CustomRecordTypeUpsert method.

Table 79. Arguments Taken by the CustomRecordTypeUpsert Method

Name	Description	Required	Default	I/O
apData:CustomRecordType	The custom record type data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CustomWebLinkRead

Extracts custom Web link configuration data.

Usage

Use the CustomWebLinkRead method to extract custom Web link configuration data for an object.

The object is specified with the <ObjectName> child element of apQuery:WebLink.

See [“CustomWebLinkUpsert” on page 231](#) for information about the custom Web link data specified in apData:WebLinkSet.

Arguments

[Table 80](#) describes the arguments taken by the CustomWebLinkRead method.

Table 80. Arguments Taken by the CustomWebLinkRead Method

Name	Description	Required	Default	I/O
apQuery:WebLinkSet	The custom Web link query criteria.	Yes	Not applicable	Input
apData:ListOfWebLinkSet	The extracted custom Web link data.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web link is returned.

CustomWebLinkReadAll

Extracts all of the custom Web link configuration data for a company.

Usage

Use the CustomWebLinkReadAll method to read all of the custom Web links.

There is no input argument; the method simply returns all of the custom Web link data.

See [“CustomWebLinkUpsert” on page 231](#) for information about the custom Web link data specified in apData:WebLinkSet.

Arguments

[Table 81](#) describes the arguments taken by the CustomWebLinkReadAll method.

Table 81. Arguments Taken by the CustomWebLinkReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfWebLinkSet	The extracted data for all custom Web links.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web links are returned.

CustomWebLinkUpsert

Updates custom Web link data or inserts new custom Web link data for an object

Usage

Use the CustomWebLinkUpsert method to insert custom Web link or update existing custom Web link data for an object.

The object is specified with the <ObjectName> child element of apData:WebLinkSet.

The apdata:WebLinkSet argument contains the following elements containing custom Web link data:

- **Name.** The name of the custom Web link field. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <WebLinkTranslation> element.
- **IntegrationTag.** The field's integration tag name.
- **DisplayText.** The label for the hyperlink of the custom Web link.
- **WebLinkTarget.** The behavior when a user clicks the hyperlink. Open in Current window, Open in Custom Tab, or Open in New window.
- **TargetCustomWebTab.** The required custom Web tab, if Open in Custom Tab is specified for WebLinkTarget.
- **RefreshParentWindow.** Whether the parent window is refreshed after a new window is opened. Only applicable if Open in New window is specified for WebLinkTarget.
- **DisplayOptions.** A character indicating the pages where the custom Web link is displayed: D for Detail Page, E for Edit Page, or B for Both.
- **ActiveLinkCondition.** The condition that controls whether the custom Web link is active on the Detail and Edit pages.
- **DisplayLinkCondition.** The condition that controls whether the custom Web link is visible on the Detail and Edit pages.
- **URL.** The URL that is invoked when the user clicks the custom Web link.
- **ListOfWebLinkTranslations.** The display names in the languages activated for your company:
 - **DisplayName.** The display name of the custom Web link in the language identified by the language code.
 - **DisplayText.** The label for the hyperlink, in the language identified by the language code.
 - **LangCode.** The three-letter language code for the language.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

For more information about custom Web links, see *Oracle CRM On Demand Online Help*.

Arguments

[Table 82](#) describes the arguments taken by the CustomWebLinkUpsert method.

Table 82. Arguments Taken by the CustomWebLinkUpsert Method

Name	Description	Required	Default	I/O
apData:WebLinkSet	The custom Web link data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

CustomWebTabDelete

Deletes a custom Web tab.

Usage

Use the CustomWebTabDelete method to delete a custom Web tab matching the input criteria.

The custom Web tab is specified with the <DisplayName> child element of apQuery:CustomWebTab.

See [“CustomWebTabUpsert” on page 234](#) for information about the custom Web tab data specified in apData:CustomWebTab.

Arguments

[Table 83](#) describes the arguments taken by the CustomWebTabDelete method.

Table 83. Arguments Taken by the CustomWebTabDelete Method

Name	Description	Required	Default	I/O
apQuery:CustomWebTab	The custom Web tab query criteria.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status of the call is returned.

CustomWebTabRead

Extracts custom Web tab configuration data.

Usage

Use the CustomWebTabRead method to extract custom Web tab configuration data matching the input criteria.

The custom Web tab is specified with the <DisplayName> child element of apQuery:CustomWebTab.

See [“CustomWebTabUpsert” on page 234](#) for information about the custom Web tab data specified in apData:CustomWebTab.

Arguments

[Table 84](#) describes the arguments taken by the CustomWebTabRead method.

Table 84. Arguments Taken by the CustomWebTabRead Method

Name	Description	Required	Default	I/O
apQuery:CustomWebTab	The custom Web tab query criteria.	Yes	Not applicable	Input
apData:CustomWebTab	The extracted custom Web tab data.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web tab is returned.

CustomWebTabReadAll

Extracts all of the custom Web tab configuration data for a company.

Usage

Use the CustomWebTabReadAll method to read all of the custom Web tabs.

There is no input argument; the method simply returns all of the custom Web tab data.

See [“CustomWebTabUpsert” on page 234](#) for information about the custom Web tab data specified in apData:CustomWebTab.

Arguments

Table 85 describes the arguments taken by the CustomWebTabReadAll method.

Table 85. Arguments Taken by the CustomWebTabReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfCustomWebTab	The extracted data for all custom Web tabs.	Not applicable	Not applicable	Output

Return Value of the Call

The custom Web tabs are returned.

CustomWebTabUpsert

Updates an existing custom Web tab or inserts a new custom Web tab.

Usage

Use the CustomWebTabUpsert method to insert a custom Web tab or update an existing custom Web tab.

The apdata:CustomWebTab argument contains the following elements containing custom Web tab data:

- **DisplayName.** The name of the custom Web tab. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <CustomWebTabTranslation> element.
- **Description.** A description of the custom Web tab.
- **Type.** The type of custom Web Tab, which can be URL or HTML.
- **HTMLHeadAdditions.** The HTML to be added within the <head> element of the custom Web tab. (For type HTML only).
- **WebTabHTML.** The HTML to be added within the <body> element of the custom Web tab. (For type HTML only).
- **Url.** The URL for the custom Web tab.
- **FrameHeight.** The frame height defined for the custom Web tab.
- **FrameWidth.** The frame width defined for the custom Web tab.
- **IconName.** The filename of the icon used for the custom Web tab.
- **ListOfCustomWebTabTranslation.** The display names in the languages activated for your company:
 - **LanguageCode.** The three-letter language code for the language.

- **DisplayName.** The display name of the custom Web tab in the language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.

Arguments

[Table 86](#) describes the arguments taken by the CustomWebTabUpsert method.

Table 86. Arguments Taken by the CustomWebTabUpsert Method

Name	Description	Required	Default	I/O
apData:CustomWebTab	The custom Web tab data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

DetailPageCustomWebAppletRead

Extracts detail page Web applet configuration data.

Usage

Use the DetailPageCustomWebAppletRead method to read configuration data for a detail page Web applet.

The record type and Web applet are specified with the <ObjectName> and <DisplayName> child elements of apQuery:DetailPageCustomWebApplet.

See [“DetailPageCustomWebAppletUpsert” on page 237](#) for information about the data specified in apData:ListOfDetailPageCustomWebApplet.

Arguments

[Table 87](#) describes the arguments taken by the `DetailPageCustomWebAppletRead` method.

Table 87. Arguments Taken by the `DetailPageCustomWebAppletRead` Method

Name	Description	Required	Default	I/O
<code>apQuery:DetailPageCustomWebApplet</code>	The Web applet query criteria.	Yes	Not applicable	Input
<code>apData:ListOfDetailPageCustomWebApplet</code>	The extracted data for the Web applet.	Not applicable	Not applicable	Output

Return Value of the Call

The Web applet data is returned.

DetailPageCustomWebAppletReadAll

Extracts all of the detail page Web applet configuration data for a company.

Usage

Use the `DetailPageCustomWebAppletReadAll` method to read all of the detail page Web applets.

There is no input argument; the method simply returns all of the Web applet data.

See [“DetailPageCustomWebAppletUpsert” on page 237](#) for information about the data specified in `apData:ListOfDetailPageCustomWebApplet`.

Arguments

[Table 88](#) describes the arguments taken by the `DetailPageCustomWebAppletReadAll` method.

Table 88. Arguments Taken by the `DetailPageCustomWebAppletReadAll` Method

Name	Description	Required	Default	I/O
<code>apData:ListOfDetailPageCustomWebApplet</code>	The extracted data for all detail page Web applets.	Not applicable	Not applicable	Output

Return Value of the Call

The detail page Web applets are returned.

DetailPageCustomWebAppletUpsert

Updates an existing detail page Web applet or inserts a new Web applet.

Usage

Use the `DetailPageCustomWebAppletUpsert` method to insert or update an existing detail page Web applet.

The `apData:DetailPageCustomWebApplet` argument contains the following elements containing detail page Web applet data:

- **ObjectName.** The record type. Required.
- **DisplayName.** The name of the Web applet. Required.
- **Description.** A description for the Web applet.
- **Height.** The height of the applet in pixels.
- **FeedType.** Settings for an RSS feed applet:
 - **URL.** The URL that is invoked when the user clicks the hyperlink.
 - **OverrideURLFlag.** Whether or not the user can override the URL.
 - **TargetWindow.** The window in which the Web applet opens: Current Window or New Window.
- **HTMLType.** Settings for an HTML Web applet:
 - **HTMLHeader.** HTML code used in an `<iframe>` element within the HTML header of the page to which you add the Web applet.
 - **HTMLBody.** HTML code used in an `<iframe>` element within the HTML body of the page to which you add the Web applet.
 - **AlwaysRun.** Whether the Web applet will be executed if the applet is minimized.
- **URLType.** Settings for a URL Web applet:
 - **URL.** The URL that is invoked when the user clicks the hyperlink.
 - **AlwaysRun.** Whether the Web applet will be executed if the applet is minimized.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets.

Arguments

[Table 89](#) describes the arguments taken by the `DetailPageCustomWebAppletUpsert` method.

Table 89. Arguments Taken by the `DetailPageCustomWebAppletUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:DetailPageCustomWebApplet</code>	The Web applet data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

FieldManagementCreate

Creates field management configuration data.

Usage

Use the `FieldManagementCreate` method to create field management configuration data.

The object is specified with the `<ObjectName>` child element of `apData:CustomField`. The field management data for the object is contained in elements similar to those of the `apData:FieldManagement` argument used by the `FieldManagementUpsert` method, see [“FieldManagementUpsert” on page 240](#).

Arguments

[Table 90](#) describes the arguments taken by the `FieldManagementCreate` method.

Table 90. Arguments Taken by the `FieldManagementCreate` Method

Name	Description	Required	Default	I/O
<code>apData:CustomField</code>	The field management query criteria.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status of the call is returned.

FieldManagementRead

Extracts field management configuration data.

Usage

Use the FieldManagementRead method to extract field management configuration data for an object.

The object is specified with the <ObjectName> child element of apQuery:FieldSet.

If the <IncludeAll> child element of apQuery:FieldSet specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

Regardless of whether only customized data is returned, or preconfigured and customized data, translated data for all languages enabled by the company are included in the output.

See [“FieldManagementUpsert” on page 240](#) for information about the field management data specified in apData:FieldSet.

By default, FieldManagementRead only extracts field definitions for fields that have been created by the user, or preconfigured fields that have been modified. Unmodified preconfigured fields are bypassed because they exist on all company instances. See the information in this section about <IncludeAll> for details on how to include all fields, including preconfigured fields, in the SOAP response.

Arguments

[Table 91](#) describes the arguments taken by the FieldManagementRead method.

Table 91. Arguments Taken by the FieldManagementRead Method

Name	Description	Required	Default	I/O
apQuery:FieldSet	The field management query criteria.	Yes	Not applicable	Input
apData:ListOfFieldSet	The extracted field management data.	Not applicable	Not applicable	Output

Return Value of the Call

The field management data is returned.

FieldManagementReadAll

Extracts all of the field management configuration data for a company.

Usage

Use the `FieldManagementReadAll` method to read all of the field management data.

If the input argument `<IncludeAll>` specifies true, the method returns all configuration data, including preconfigured and customized data. If `<IncludeAll>` specifies false, or is not included in the request, the method returns customized configuration data only.

Regardless of whether only customized data is returned, or preconfigured and customized data, translated data for all languages enabled by the company are included in the output.

See [“FieldManagementUpsert” on page 240](#) for information about the field management data specified in `apData:FieldSet`.

Arguments

[Table 92](#) describes the arguments taken by the `FieldManagementReadAll` method.

Table 92. Arguments Taken by the `FieldManagementReadAll` Method

Name	Description	Required	Default	I/O
<code>IncludeAll</code>	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
<code>apData:ListOfFieldSet</code>	The extracted field management data.	Not applicable	Not applicable	Output

Return Value of the Call

The field management data is returned.

FieldManagementUpsert

Updates existing field management data or inserts field management data for an object.

Usage

Use the `FieldManagementUpsert` method to insert field management data or update existing field management data for an object.

The object is specified with the `<ObjectName>` child element of `apData:FieldSet`.

The `apdata:FieldSet` argument contains the following elements containing field management data:

- **Name.** The system name of the field.
- **DisplayName.** The name of the field used in the UI. When performing an insert operation, this name is used as the display name, if a display name is not provided in the `<FieldManagementTranslation>` element.

- **FieldType.** The type of field.
- **IntegrationTag.** The field's custom integration tag name. This tag can be modified by the user.
- **GenericIntegrationTag.** The field's default integration tag name assigned by Oracle CRM On Demand. This tag cannot be modified by the user. The default integration tag name has the format *fieldtypeDisplay_Name*. For example, stMy_Custom_Field would be assigned to a custom field of type Text (Short) with the display name of My Custom Field.

NOTE: For noncustom fields, the <GenericIntegrationTag> and <IntegrationTag> values are the same.
- **Required.** Whether the field is a required field (true), or is not a required field (false).
- **ReadOnly.** Whether the field is a read-only field (true), or is not a read-only field (false).
- **DefaultValue.** The default value of the field.
- **FieldValidation.** The expression for the field validation rules for the field.
- **PostDefault.** Whether the field is prepopulated with the default value when a user creates a new record and the record is saved.
- **ValidationErrorMsg.** The custom error message displayed if the field fails field validation.
- **ListOfFieldTranslations.** The field management data in the languages activated for your company:
 - **LanguageCode.** The three-letter language code for the language.
 - **DisplayName.** The display name of the field in the language identified by the language code.
 - **ValidationErrorMsg.** The custom validation error message in the language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

For more information about field management, including field validation, see *Oracle CRM On Demand Online Help*.

Arguments

[Table 93](#) describes the arguments taken by the FieldManagementUpsert method.

Table 93. Arguments Taken by the FieldManagementUpsert Method

Name	Description	Required	Default	I/O
apData:FieldSet	The field management data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

HomepageLayoutReadAll

Extracts all of the homepage layout configuration data for a company.

Usage

Use the HomepageLayoutReadAll method to read all of the homepage layouts.

There is no input argument; the method simply returns all of the homepage layout data.

See [“HomepageLayoutUpsert” on page 243](#) for information about the homepage layout data specified in apData:ListOfHomepageLayout.

Arguments

[Table 94](#) describes the arguments taken by the HomepageLayoutReadAll method.

Table 94. Arguments Taken by the HomepageLayoutReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfHomepageLayout	The extracted data for all homepage layouts.	Not applicable	Not applicable	Output

Return Value of the Call

The homepage layouts are returned.

HomepageLayoutUpsert

Updates an existing homepage layout or inserts a new homepage layout.

Usage

Use the `HomepageLayoutUpsert` method to insert or update an existing homepage layout.

The `apdata:HomepageLayout` argument contains the following elements containing homepage layout data:

- **ObjectName.** The name of the object.
- **LayoutName.** The name of the homepage layout.
- **Description.** A description of the homepage layout.
- **ListOfAvailableSection.** Contains all of the available sections for the homepage layout. The data for a particular section is contained in the following elements:
 - **SectionName.** The name of the section.
 - **SectionLayoutName.** The name of the list of available sections, such as Available Sections.
- **ListOfSelectedSection.** Contains all of the selected sections for the homepage layout. The data for a particular section is contained in the following elements:
 - **SectionName.** The name of the section.
 - **SectionLayoutName.** The name of the lists of sections, such as Left Side and Right Side.
 - **Sequence.** The sequence of the section within the lists of sections:
 - A sequence value of -2 displays the section in the Available column.
 - All even numbers, including 0 (zero) displays the specified section in the Left Side column, where 0 (zero) represents the first section, and so on.
 - All odd numbers display the section in the Right Side column, where a value of 1 represents the first section, and so on.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of section applet names to section names used in the Oracle CRM On Demand user interface.
- Mappings of section layout names to section names used in the Oracle CRM On Demand user interface.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

[Table 95](#) describes the arguments taken by the HomepageLayoutUpsert method.

Table 95. Arguments Taken by the HomepageLayoutUpsert Method

Name	Description	Required	Default	I/O
apData:HomepageLayout	The homepage layout data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

HomepageCustomWebAppletRead

Extracts homepage Web applet configuration data.

Usage

Use the HomepageCustomWebAppletRead method to read configuration data for a homepage Web applet.

The record type and Web applet are specified with the <ObjectName> and <DisplayName> child elements of apQuery:HomePageCustomWebApplet.

See [“HomepageCustomWebAppletUpsert” on page 245](#) for information about the data specified in apData:ListOfHomePageCustomWebApplet.

Arguments

[Table 96](#) describes the arguments taken by the HomepageCustomWebAppletRead method.

Table 96. Arguments Taken by the HomepageCustomWebAppletRead Method

Name	Description	Required	Default	I/O
apQuery:HomePageCustomWebApplet	The Web applet query criteria.	Yes	Not applicable	Input
apData:ListOfHomePageCustomWebApplet	The extracted data for the Web applet.	Not applicable	Not applicable	Output

Return Value of the Call

The Web applet data is returned.

HomepageCustomWebAppletReadAll

Extracts all of the homepage Web applet configuration data for a company.

Usage

Use the HomepageCustomWebAppletReadAll method to read all of the homepage Web applets.

There is no input argument; the method simply returns all of the Web applet data.

See [“HomepageCustomWebAppletUpsert” on page 245](#) for information about the data specified in apData:ListOfHomePageCustomWebApplet.

Arguments

[Table 97](#) describes the arguments taken by the HomepageCustomWebAppletReadAll method.

Table 97. Arguments Taken by the HomepageCustomWebAppletReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfHomePageCustomWebApplet	The extracted data for all homepage Web applets.	Not applicable	Not applicable	Output

Return Value of the Call

The homepage Web applets are returned.

HomepageCustomWebAppletUpsert

Updates an existing homepage Web applet or inserts a new Web applet.

Usage

Use the HomepageCustomWebAppletUpsert method to insert or update an existing homepage Web applet or a custom homepage report.

The apData:HomePageCustomWebApplet argument contains the following elements containing homepage Web applet data:

- **ObjectName.** The record type. Required.
- **DisplayName.** The name of the Web applet. Required.

- **Description.** A description for the Web applet.
- **Width.** The column width of the applet: 1 for Single, 2 for Double.
- **Height.** The height of the applet in pixels.
- **FeedType.** Settings for an RSS feed applet:
 - **URL.** The URL that is invoked when the user clicks the hyperlink.
 - **OverrideURLFlag.** Whether or not the user can override the URL.
 - **TargetWindow.** The window in which the Web applet opens: Current Window or New Window.
- **HTMLType.** Settings for an HTML Web applet:
 - **HTMLHeader.** HTML code used in an <iframe> element within the HTML header of the page to which you add the Web applet.
 - **HTMLBody.** HTML code used in an <iframe> element within the HTML body of the page to which you add the Web applet.
- **URLType.** Settings for a URL Web applet:
 - **URL.** The URL that is invoked when the user clicks the hyperlink.
- **ReportType.** Settings for a custom homepage report:
 - **ReportPath.** The path of the report.
 - **ReportFlag.** Whether the report will run automatically, and users will not have to click a link to update the report.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets and custom homepage reports.

Arguments

[Table 98](#) describes the arguments taken by the `HomepageCustomWebAppletUpsert` method.

Table 98. Arguments Taken by the `HomepageCustomWebAppletUpsert` Method

Name	Description	Required	Default	I/O
apData:HomePageCustomWebApplet	The Web applet data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

HomeTabCustomWebAppletRead

Extracts My Homepage tab Web applet configuration data.

Usage

Use the HomeTabCustomWebAppletRead method to read configuration data for a My Homepage tab Web applet.

The Web applet is specified with the <DisplayName> child element of apQuery:HomeTabCustomWebApplet.

See [“HomeTabCustomWebAppletUpsert” on page 248](#) for information about the data specified in apData:ListOfHomeTabCustomWebApplet.

Arguments

[Table 99](#) describes the arguments taken by the HomeTabCustomWebAppletRead method.

Table 99. Arguments Taken by the HomeTabCustomWebAppletRead Method

Name	Description	Required	Default	I/O
apQuery:HomeTabCustomWebApplet	The Web applet query criteria.	Yes	Not applicable	Input
apData:ListOfHomeTabCustomWebApplet	The extracted data for the Web applet.	Not applicable	Not applicable	Output

Return Value of the Call

The Web applet data is returned.

HomeTabCustomWebAppletReadAll

Extracts all of the My Homepage tab Web applet configuration data for a company.

Usage

Use the HomeTabCustomWebAppletReadAll method to read all of the My Homepage tab Web applets.

There is no input argument; the method simply returns all of the Web applet data.

See [“HomeTabCustomWebAppletUpsert” on page 248](#) for information about the data specified in apData:ListOfHomeTabCustomWebApplet.

Arguments

Table 100 describes the arguments taken by the HomeTabCustomWebAppletReadAll method.

Table 100. Arguments Taken by the HomeTabCustomWebAppletReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfHomeTabCustomWebApplet	The extracted data for all My Homepage tab Web applets.	Not applicable	Not applicable	Output

Return Value of the Call

The My Homepage tab Web applets are returned.

HomeTabCustomWebAppletUpsert

Updates an existing My Homepage tab Web applet or inserts a new Web applet.

Usage

Use the HomeTabCustomWebAppletUpsert method to insert or update an existing My Homepage tab Web applet or custom report.

The apData:HomeTabCustomWebApplet argument contains the following elements containing My Homepage tab Web applet data:

- **DisplayName.** The name of the Web applet. Required.
- **Description.** A description for the Web applet.
- **Width.** The column width of the applet: 1 for Single, 2 for Double.
- **Height.** The height of the applet in pixels.
- **FeedType.** Settings for an RSS feed applet:
 - **URL.** The URL that is invoked when the user clicks the hyperlink.
 - **OverrideURLFlag.** Whether or not the user can override the URL.
 - **TargetWindow.** The window in which the Web applet opens: Current Window or New Window.
- **HTMLType.** Settings for an HTML Web applet:
 - **HTMLHeader.** HTML code used in an <iframe> element within the HTML header of the page to which you add the Web applet.
 - **HTMLBody.** HTML code used in an <iframe> element within the HTML body of the page to which you add the Web applet.
- **URLType.** Settings for a URL Web applet:

- **URL.** The URL that is invoked when the user clicks the hyperlink.
- **ReportType.** Settings for a custom report.
 - **ReportPath.** The path for the report.
 - **ReportFlag.** Whether the report will run automatically, and users will not have to click a link to update the report.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets and custom reports.

Arguments

[Table 101](#) describes the arguments taken by the HomeTabCustomWebAppletUpsert method.

Table 101. Arguments Taken by the HomeTabCustomWebAppletUpsert Method

Name	Description	Required	Default	I/O
apData:HomeTabCustomWebApplet	The Web applet data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

IndustryRead

Extracts industry configuration data.

Usage

Use the IndustryRead method to extract industry configuration data for your company.

The industry is specified with the <Name> child element of apQuery:Industry.

See [“IndustryReadAll” on page 250](#) for information about the industry data specified in apData:Industry.

Arguments

Table 102 describes the arguments taken by the IndustryRead method.

Table 102. Arguments Taken by the IndustryRead Method

Name	Description	Required	Default	I/O
apQuery:Industry	The industry query criteria.	Yes	Not applicable	Input
apData:ListOfIndustry	The extracted industry data.	Not applicable	Not applicable	Output

Return Value of the Call

The industry data is returned.

IndustryReadAll

Extracts all of the industry configuration data for a company.

Usage

Use the IndustryReadAll method to read all of the industry configuration data for your company.

There is no input argument; the method simply returns all of the industry data.

The apdata:Industry argument contains the following elements containing industry data:

- **Name.** The name of the industry.
- **Active.** Whether the industry is enabled.
- **Type.** The type of industry classification used, for example, 4-digit SIC.
- **DisplayName.** The display name of the industry.
- **SICode.** The code for the industry according to the Standard Industrial Classification (SIC) system.
- **MarkforTranslation.** Whether the industry display name is translated.
- **ListOfIndustryTranslation.** The industry names in the languages activated for your company:
 - **LanguageCode.** The three-letter language code for the language.
 - **Title.** The display name of the industry in the language identified by the language code.

Arguments

[Table 103](#) describes the arguments taken by the `IndustryReadAll` method.

Table 103. Arguments Taken by the `IndustryReadAll` Method

Name	Description	Required	Default	I/O
<code>apData:ListOfIndustry</code>	The extracted industry data for the company.	Not applicable	Not applicable	Output

Return Value of the Call

The industry data for the company is returned.

PageLayoutFieldRead

Extracts all of the field configuration data for a detail page of an object.

Usage

Use the `PageLayoutFieldRead` method to read the field data for an object's detail page.

The object is specified with the `<ObjectName>` child element of `apQuery:PageLayout`.

If the `<CurrentUserOnly>` child element of `apQuery:PageLayout` specifies `true`, page layout field data for the current user only is returned.

If the `<IncludeAll>` child element of `apQuery:PageLayout` specifies `true`, the method returns configuration data including preconfigured and customized data. If `<IncludeAll>` specifies `false`, or is not included in the request, the method returns customized configuration data only.

See [“PageLayoutFieldUpsert” on page 253](#) for information about the page layout field data specified in `apData:PageLayout`.

Arguments

[Table 104](#) describes the arguments taken by the `PageLayoutFieldRead` method.

Table 104. Arguments Taken by the `PageLayoutFieldRead` Method

Name	Description	Required	Default	I/O
<code>apQuery:PageLayout</code>	The page layout query criteria.	Yes	Not applicable	Input
<code>apData:ListOfPageLayout</code>	The extracted page layout field data.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout fields are returned.

PageLayoutFieldReadAll

Extracts all of the field configuration data for detail pages for a company.

Usage

Use the PageLayoutFieldReadAll method to read all of the field data for detail pages.

If the input argument <CurrentUserOnly> specifies true, all of the page layout field data for the current user only is returned.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

There is no input argument; the method simply returns all of the page layout field data.

See [“PageLayoutFieldUpsert” on page 253](#) for information about the page layout field data specified in apData:PageLayout.

Arguments

[Table 105](#) describes the arguments taken by the PageLayoutFieldReadAll method.

Table 105. Arguments Taken by the PageLayoutFieldReadAll Method

Name	Description	Required	Default	I/O
CurrentUserOnly	If true, extracts data for the current user only.	No	false	Input
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfPageLayout	The extracted page layout field data.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout fields are returned.

PageLayoutFieldUpsert

Updates existing page layout field data or inserts new page layout field data.

Usage

Use the PageLayoutFieldUpsert method to insert or update field data for a detail page layout.

The apdata:PageLayout argument contains the following elements containing page layout field data:

- **ObjectName.** The name of the object.
- **LayoutName.** The name of the page layout.
- **Description.** A description of the page layout.
- **ListOfPageLayoutField.** Contains all of the fields for the page layout. The data for a particular field is contained in the following child elements of <PageLayoutField>:
 - **FieldName.** The name of the field.
 - **ReadOnly.** Whether the field is read-only.
 - **Required.** Whether the field is required.
 - **Section.** An integer identifying the section in which the field appears. The value 0 represents the first section, 1 the next section, and so on.
 - **Column.** The column within the section in which the field appears. The value 0 represents the left-most column.
 - **Row.** The row within the section in which the field appears. The value 0 represents the first row within a section and column, 1 the next row, and so on.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

Table 106 describes the arguments taken by the PageLayoutFieldUpsert method.

Table 106. Arguments Taken by the PageLayoutFieldUpsert Method

Name	Description	Required	Default	I/O
apData:PageLayout	The page layout field data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

PageLayoutRelatedInformationRead

Extracts all of the related information for a detail page layout for an object.

Usage

Use the `PageLayoutRelatedInformationRead` method to read all of the related information data for an object.

The object is specified with the `<ObjectName>` child element of `apQuery:PageLayoutRelatedInformation`.

If the `<CurrentUserOnly>` child element of `apQuery:PageLayoutRelatedInformation` specifies `true`, related information data for the current user only is returned.

If the `<IncludeAll>` child element of `apQuery:PageLayoutRelatedInformations` specifies `true`, the method returns configuration data including preconfigured and customized data. If `<IncludeAll>` specifies `false`, or is not included in the request, the method returns customized configuration data only.

See [“PageLayoutRelatedInformationUpsert” on page 255](#) for information about the page layout related information specified in `apData:PageLayoutRelatedInformation`.

Arguments

[Table 107](#) describes the arguments taken by the `PageLayoutRelatedInformationRead` method.

Table 107. Arguments Taken by the `PageLayoutRelatedInformationRead` Method

Name	Description	Required	Default	I/O
<code>apQuery:PageLayoutRelatedInformation</code>	The related information query criteria.	Yes	Not applicable	Input
<code>apData:ListOfPageLayoutRelatedInformation</code>	The extracted page layout related information.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout related information is returned.

PageLayoutRelatedInformationReadAll

Extracts all of the related information for detail page layouts for a company.

Usage

Use the `PageLayoutRelatedInformationReadAll` method to read all of the related information data for detail page layouts.

If the input argument `<CurrentUserOnly>` is true, all of the related information data for the current user only is returned.

If the input argument `<IncludeAll>` specifies true, the method returns all configuration data, including preconfigured and customized data. If `<IncludeAll>` specifies false, or is not included in the request, the method returns customized configuration data only.

See [“PageLayoutRelatedInformationUpsert” on page 255](#) for information about the page layout related information specified in `apData:PageLayoutRelatedInformation`.

Arguments

[Table 108](#) describes the arguments taken by the `PageLayoutRelatedInformationReadAll` method.

Table 108. Arguments Taken by the `PageLayoutRelatedInformationReadAll` Method

Name	Description	Required	Default	I/O
<code>CurrentUserOnly</code>	If true, extracts data for the current user only	No	false	Input
<code>IncludeAll</code>	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
<code>apData:ListOfPageLayoutRelatedInformation</code>	The extracted page layout related information.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout related information is returned.

PageLayoutRelatedInformationUpsert

Updates existing page layout related information or inserts new page layout related information.

Usage

Use the `PageLayoutRelatedInformationUpsert` method to insert or update related information for a detail page layout.

The `apdata:PageLayoutRelatedInformation` argument contains the following elements containing related information for a page layout:

- **ObjectName.** The name of the object.
- **LayoutName.** The name of the page layout.
- **ListOfPageRelatedInformation.** Contains all of the related information for the page layout. The data for a particular related information section is contained in the following child elements of <PageRelatedInformation>:
 - **Name.** A child object ID, equivalent to the name of a related information section. See *Oracle Migration Tool On Demand Guide* for reference tables about the mapping of child object IDs to UI display names for the child objects.
 - **Availability.** An integer with the range of values -1 through 1, specifying in which list the related information appears in the page layout: Not Available, Available, or Displayed respectively.

For values of 1 (Displayed), the order of the <PageRelatedInformation> elements determines the order in which the related information sections are listed in the layout in the UI.
 - **RelatedInformationLayout.** The name of the corresponding related information layout.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

Table 109 describes the arguments taken by the `PageLayoutRelatedInformationUpsert` method.

Table 109. Arguments Taken by the `PageLayoutRelatedInformationUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:PageLayoutRelatedInformation</code>	The page layout related information to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

PageLayoutSectionReadAll

Extracts all of the section data for detail page layouts for a company.

Usage

Use the `PageLayoutSectionReadAll` method to read all of the section data for detail page layouts.

If the input argument `<CurrentUserOnly>` is true, all of the related information data for the current user only is returned.

If the input argument `<IncludeAll>` specifies true, the method returns all configuration data, including preconfigured and customized data. If `<IncludeAll>` specifies false, or is not included in the request, the method returns customized configuration data only.

See [“PageLayoutSectionUpsert” on page 257](#) for information about the page layout section data specified in `apData:PageLayoutFormSection`.

Arguments

[Table 110](#) describes the arguments taken by the `PageLayoutSectionReadAll` method.

Table 110. Arguments Taken by the `PageLayoutSectionReadAll` Method

Name	Description	Required	Default	I/O
<code>apData:ListOfPageLayoutFormSection</code>	The extracted page layout section data.	Not applicable	Not applicable	Output

Return Value of the Call

The page layout section data are returned.

PageLayoutSectionUpsert

Updates existing page layout section data or inserts new page layout section data.

Usage

Use the `PageLayoutSectionUpsert` method to insert or update section data for a detail page layout.

The `apdata:PageLayoutFormSection` argument contains the following elements containing page layout section data:

- **ObjectName.** The name of the object.
- **LayoutName.** The name of the page layout.
- **ListOfPageFormSection.** The section data for each section. The data is contained in the following child elements of `<PageFormSection>`:
 - **SectionSequence.** The position in which the section appears in the page layout.

- **ListOfFormSectionTranslation.** The section names in the languages activated for your company. The data for a particular section is contained in the following child elements of `<FormSectionTranslation>`:
 - ❑ **LangId.** The three-letter language code for the language.
 - ❑ **DisplayName.** The display name of the section in the language identified by the language code.

See the *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.

Arguments

[Table 111](#) describes the arguments taken by the `PageLayoutSectionUpsert` method.

Table 111. Arguments Taken by the `PageLayoutSectionUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:PageLayoutFormSection</code>	The page layout section data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

PicklistRead

Extracts picklist configuration data.

Usage

Use the `PicklistRead` method to extract picklist configuration data for an object.

The object is specified with the `<ObjectName>` child element of `apQuery:PicklistSet`.

If the `<IncludeAll>` child element of `apQuery:PicklistSet` specifies `true`, the method returns configuration data including preconfigured and customized data. If `<IncludeAll>` specifies `false`, or is not included in the request, the method returns customized configuration data only.

See [“PicklistUpsert” on page 260](#) for information about the picklist data specified in `apData:PicklistSet`.

Arguments

[Table 112](#) describes the arguments taken by the PicklistRead method.

Table 112. Arguments Taken by the PicklistRead Method

Name	Description	Required	Default	I/O
apQuery:PicklistSet	The picklist query criteria.	Yes	Not applicable	Input
apData:ListOfPicklistSet	The extracted picklist data.	Not applicable	Not applicable	Output

Return Value of the Call

The picklists for the object are returned.

PicklistReadAll

Extracts all of the picklist configuration data for a company.

Usage

Use the PicklistReadAll method to read all of the picklists.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See [“PicklistUpsert” on page 260](#) for information about the picklist data specified in apData:PicklistSet.

Arguments

[Table 113](#) describes the arguments taken by the PicklistReadAll method.

Table 113. Arguments Taken by the PicklistReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfPicklistSet	The extracted data for all picklists for all objects.	Not applicable	Not applicable	Output

Return Value of the Call

The picklists are returned.

PicklistUpsert

Updates an existing set of picklists or inserts a new set of picklists.

Usage

Use the `PicklistUpsert` method to insert a set of picklists or update an existing set of picklists for an object.

The object is specified with the `<ObjectName>` element of `apdata:PicklistSet`.

The `apdata:PicklistSet` argument contains the following elements containing picklist data:

- **Name.** The name of the picklist. When performing an insert operation, this name is used as the display name, if a display name is not provided in the `<PicklistTranslation>` element.
- **PickListValue.** The picklist value.
- **ValueId.** The number that determines the position of the value in the picklist.
- **Disabled.** Whether the picklist value is disabled.
- **ListOfPicklistTranslation.** The picklist value in the languages activated for your company:
 - **LanguageCode.** The three-letter language code for the language.
 - **Value.** The picklist value in the language identified by the language code.
 - **Order.** The order number of the picklist value according to language identified by the language code.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

[Table 114](#) describes the arguments taken by the PicklistUpsert method.

Table 114. Arguments Taken by the PicklistUpsert Method

Name	Description	Required	Default	I/O
apData:PicklistSet	The picklist data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

RelatedInformationLayoutRead

Extracts all of the related information layout data for an object.

Usage

Use the RelatedInformationLayoutRead method to read all of the related information layout data for an object.

The object is specified with the <ObjectName> child element of apQuery:RelatedInformationLayout.

If the <CurrentUserOnly> child element of apQuery:RelatedInformationLayout specifies true, related information layout data for the current user only is returned.

If the <IncludeAll> child element of apQuery:RelatedInformationLayout specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See [“RelatedInformationLayoutUpsert” on page 263](#) for information about the related information layout data specified in apData:RelatedInformationLayout.

Arguments

[Table 115](#) describes the arguments taken by the RelatedInformationLayoutRead method.

Table 115. Arguments Taken by the RelatedInformationLayoutRead Method

Name	Description	Required	Default	I/O
apQuery:RelatedInformationLayout	The related information layout query criteria.	Yes	Not applicable	Input
apData:ListOfRelatedInformationLayout	The extracted related information layout data.	Not applicable	Not applicable	Output

Return Value of the Call

The related information layout data is returned.

RelatedInformationLayoutReadAll

Extracts all of the related information layout data for a company.

Usage

Use the RelatedInformationLayoutReadAll method to read all of the related information layout data for a company.

If the input argument <CurrentUserOnly> is true, all of the related information layout data for the current user only is returned.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See [“RelatedInformationLayoutUpsert” on page 263](#) for information about the related information data specified in apData:RelatedInformationLayout.

Arguments

Table 116 describes the arguments taken by the RelatedInformationLayoutReadAll method.

Table 116. Arguments Taken by the RelatedInformationLayoutReadAll Method

Name	Description	Required	Default	I/O
CurrentUserOnly	If true, extracts data for the current user only	No	false	Input
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfRelatedInformationLayout	The extracted related information layout data.	Not applicable	Not applicable	Output

Return Value of the Call

The related information layout data is returned.

RelatedInformationLayoutUpsert

Updates existing related information layout data or inserts new related information layout data.

Usage

Use the RelatedInformationLayoutUpsert method to insert or update related information layout data for an object.

The apdata:RelatedInformationLayout argument contains the following elements containing related information layout data:

- **ObjectName.** The name of the object.
- **RelatedInformationName.** A child object ID, equivalent to the name of a related information section. See *Oracle Migration Tool On Demand Guide* for reference tables about the mapping of child object IDs to UI display names for the child objects.
- **Name.** The name of the related information layout.
- **Description.** A description of the related information layout.
- **ListOfSelectedField.** Contains all of the selected fields for the related information layouts. The field data for a particular related information layout is contained in the following child elements of <SelectedFieldData>:
 - **FieldName.** The name of the selected field. Required.

- **Sequence.** An integer with the range of values 1 through 9, specifying the sequence of the field in the Selected Fields list.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

Arguments

[Table 117](#) describes the arguments taken by the RelatedInformationLayoutUpsert method.

Table 117. Arguments Taken by the RelatedInformationLayoutUpsert Method

Name	Description	Required	Default	I/O
apData:RelatedInformationLayout	The related information layout to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

RoleRead

Extracts role configuration data.

Usage

Use the RoleRead method to extract role configuration data.

The role is specified with the <RoleName> child element of apQuery:Role.

See [“RoleUpsert” on page 265](#) for information about the role data specified in apData:Role.

Arguments

[Table 118](#) describes the arguments taken by the RoleRead method.

Table 118. Arguments Taken by the RoleRead Method

Name	Description	Required	Default	I/O
apQuery:Role	The role query criteria.	Yes	Not applicable	Input
apData:ListOfRole	The extracted role data.	Not applicable	Not applicable	Output

Return Value of the Call

The roles for the object are returned.

RoleReadAll

Extracts all of the role configuration data for a company.

Usage

Use the RoleReadAll method to read all of the roles.

See [“RoleUpsert” on page 265](#) for information about the role data specified in apData:Role.

Arguments

[Table 119](#) describes the arguments taken by the RoleReadAll method.

Table 119. Arguments Taken by the RoleReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfRole	The extracted data for all roles for all objects.	Not applicable	Not applicable	Output

Return Value of the Call

The roles are returned.

RoleUpsert

Updates an existing role or inserts a new role.

Usage

Use the RoleUpsert method to insert a role or update an existing role.

The role is specified with the <RoleName> element of apdata:Role.

The apdata:Role argument contains the following elements containing role data:

- **RoleName**. The name of the role. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <RoleTranslation> element.
- **Description**. The role description.
- **DefaultSalesProcess**. The default sales process for new opportunities created by users who are assigned this role.
- **ThemeName**. The default theme for the role.
- **LeadConversionLayout**. The lead conversion layout for the users assigned this role.
- **ActionBarLayout**. The action bar layout for the users assigned this role.
- **RelatedInfoFormat**. The format for related information sections on record Detail pages, which can be shown as lists or as tabs.
- **ListOfRoleTranslation**. The role name in the languages activated for your company:
 - **LanguageCode**. The three-letter language code for the language.
 - **RoleName**. The role name in the language identified by the language code.
- **ListOfRecordTypeAccess**. The record type access data:
 - **RecordName**. The record type.
 - **HasAccess**. Whether users assigned this role have access to records of this type.
 - **CanCreate**. Whether users assigned this role can create records of this type.
 - **CanReadAll**. Whether users assigned this role can see all records of this type.
- **AccessProfile**. The access profiles associated with the role:
 - **DefaultAccessProfile**. The default access profile.
 - **OwnerAccessProfile**. The owner access profile.
- **ListOfPrivilege**. The privileges associated with the role:
 - **PrivilegeName**. A privilege ID, equivalent to the name of a privilege in the Oracle CRM On Demand application. See *Oracle Migration Tool On Demand Guide* for reference tables about the mapping of privilege IDs to UI display names for the privileges.
 - **Enabled**. Whether the privilege is assigned to the role.
- **TabAccessandOrder**. The tabs available and selected for the role:
 - **ListOfAvailableTab**. The available tabs:
 - **AvailableTab**. The name of the tab.
 - **ListOfSelectedTab**. The selected tabs:
 - **TabName**. The name of the tab.

- **Order.** The position in which the tab name appears in the Selected Tabs list. A value of 0 (zero) corresponds to the first tab in the Selected Tabs list.
- **ListOfPageLayoutAssignment.** The page layout assignment for each record type:
 - **PageViewType.** The page view type: Static or Dynamic.
 - **RecordType.** The record type.
 - **LayoutName.** The name of the page layout. If no name is specified, the default layout is applied.
- **ListOfSearchLayoutAssignment.** The search layout assignment for each record type:
 - **RecordType.** The record type.
 - **LayoutName.** The name of the search layout.
- **ListOfHomepageLayoutAssignment.** The homepage layout assignment for each record type:
 - **RecordType.** The record type.
 - **LayoutName.** The name of the homepage layout.

See *Oracle Migration Tool On Demand Guide* for reference tables about the following:

- Mappings of three-letter language codes to languages.
- Information on privilege IDs, and their corresponding names, and descriptions.

Arguments

[Table 120](#) describes the arguments taken by the RoleUpsert method.

Table 120. Arguments Taken by the RoleUpsert Method

Name	Description	Required	Default	I/O
apData:Role	The role data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

SalesAssessmentTemplateRead

Extracts sales assessment template configuration data.

Usage

Use the `SalesAssessmentTemplateRead` method to extract sales assessment template configuration data matching the input criteria.

The sales assessment template is specified with the `<Name>` child element of `apQuery:SalesAssessmentTemplate`.

See [“SalesAssessmentTemplateUpsert” on page 269](#) for information about the sales assessment template data specified in `apData:SalesAssessmentTemplate`.

Arguments

[Table 121](#) describes the arguments taken by the `SalesAssessmentTemplateRead` method.

Table 121. Arguments Taken by the `SalesAssessmentTemplateRead` Method

Name	Description	Required	Default	I/O
<code>apQuery:SalesAssessmentTemplate</code>	The sales assessment template query criteria.	Yes	Not applicable	Input
<code>apData:ListOfSalesAssessmentTemplate</code>	The extracted sales assessment template data.	Not applicable	Not applicable	Output

Return Value of the Call

The sales assessment template is returned.

SalesAssessmentTemplateReadAll

Extracts all of the sales assessment template configuration data for a company.

Usage

Use the `SalesAssessmentTemplateReadAll` method to read all of the sales assessment templates.

See [“SalesAssessmentTemplateUpsert” on page 269](#) for information about the sales assessment template data specified in `apData:SalesAssessmentTemplate`. A number of fields that are used in the calculation of assessment scores and comparisons to thresholds are also returned.

Arguments

Table 122 describes the arguments taken by the SalesAssessmentTemplateReadAll method.

Table 122. Arguments Taken by the SalesAssessmentTemplateReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfSalesAssessmentTemplate	The extracted data for all sales assessment templates.	Not applicable	Not applicable	Output

Return Value of the Call

The sales assessment templates are returned.

SalesAssessmentTemplateUpsert

Updates an existing sales assessment template or inserts a new sales assessment template.

Usage

Use the SalesAssessmentTemplateUpsert method to insert a sales assessment template or update an existing sales assessment template.

The apdata:SalesAssessmentTemplate argument contains the following elements containing sales assessment template data:

- **Name.** The name for the sales assessment template (required).
- **Active.** Whether the sales assessment template is active.
- **Filtern.** The filters used to identify which assessment script is the appropriate one to present to a user for a task.
- **Description.** A description for the sales assessment template.
- **FieldtoMapScoreTo.** The field that displays the score in the parent record.
- **ResponseControl.** The control used for adding responses to assessment questions. Possible values are: Drop Down, Radio Button.
- **FieldtoMapOutcomeValueTo.** The field that displays the outcome value in the parent record.
- **RemoveCommentBox.** Whether the comment box is removed when the user is completing the assessment.
- **OutcomeValueIfThresholdMet.** The outcome value to display if the threshold score is met.
- **OutcomeValueIfThresholdNotMet.** The outcome value to display if the threshold score is not met.
- **ThresholdScore.** The score used to calculate the outcome of the script. If the score is equal to or higher than the threshold, the outcome is met.

- **Type.** The type of assessment script (required). Possible values are:
 - Activity Assessment
 - Account Survey
 - Business Plan Assessment
 - Contact Script
 - Lead Qualification
 - Objective Assessment
 - Opportunity Assessment
 - Service Request - Script
 - Service Request - Survey
- **ListOfSalesAssessmentTemplateAttribute.** The attributes for the sales assessment template. The following are child elements of the SalesAssessmentTemplateAttributeData element.
 - **Question.** A question to be answered by the user (required).
 - **CriteriaName.** A descriptive name for the question (required).
 - **AnswerMapToField.** The field on the parent record for displaying the answer to the question.
 - **Order.** The position in the sequence of the questions presented to the user.
 - **Weight.** A percentage value indicating the importance of the question. The sum of the weights for all questions must equal 100 (required).
 - **ListOfSalesAssessmentTemplateAttributeValue.** The attribute values of the sales assessment template. The following are child elements of the element SalesAssessmentAttributeValueData:
 - ❑ **Order.** The position in the sequence of the answers presented to the user when completing the assessment.
 - ❑ **Answer.** An answer for the question (required).
 - ❑ **Score.** The score for the answer (required).

Arguments

[Table 123](#) describes the arguments taken by the SalesAssessmentTemplateUpsert method.

Table 123. Arguments Taken by the SalesAssessmentTemplateUpsert Method

Name	Description	Required	Default	I/O
apData:SalesAssessmentTemplate	The sales assessment template data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

WorkflowActionSetRead

Extracts workflow action configuration data.

Usage

Use the WorkflowActionSetRead method to extract a workflow action set associated with a workflow rule.

The workflow rule is specified with the <WorkflowName> child element of apQuery:WorkflowActionsSet.

See [“WorkflowActionSetUpsert” on page 273](#) for information about the workflow action data specified in apQuery:WorkflowActionSet.

Arguments

[Table 124](#) describes the arguments taken by the WorkflowActionSetRead method.

Table 124. Arguments Taken by the WorkflowActionSetRead Method

Name	Description	Required	Default	I/O
apQuery:WorkflowActionSet	The workflow action query criteria.	Yes	Not applicable	Input
apData:ListOfWorkflowActionSet	The extracted workflow action data.	Not applicable	Not applicable	Output

Return Value of the Call

The workflow action data is returned.

WorkflowActionSetReadForRuleSet

Extracts workflow action configuration data for a workflow rule set.

Usage

Use the `WorkflowActionSetReadForRuleSet` method to extract all the workflow action sets for a workflow rule set for a particular record type and trigger event.

The workflow rule set is specified with the following element of `<WorkflowActionForRuleSet>`.

- **RecordType**. The record type (required).
- **TriggerEvent**. The event that triggers the workflow rule (required). The possible values and the associated trigger events are as follows:

Trigger Event	TriggerEvent LIC
When New Record Saved	WriteRecordNew
When Modified Record Saved	WriteRecordUpdated
Before Modified Record Saved	PreWriteRecord
Before Record Is Deleted	PreDeleteRecord
After Association With Parent	Associate
After Dissociation From Parent	Dissociate
When Record Is restored	RestoreRecord
When Records are merged	MergeRecords

- **ParentRecordType**. The parent record type. This is specified if the trigger event is After Association With Parent or After Dissociation From Parent. These trigger events are supported only for the Account, Contact, and Opportunity record types.

See [“WorkflowActionSetUpsert” on page 273](#) for information about the workflow action data specified in `apQuery:WorkflowActionSet`.

Arguments

Table 125 describes the arguments taken by the WorkflowActionSetReadForRuleSet method.

Table 125. Arguments Taken by the WorkflowActionSetReadForRuleSet Method

Name	Description	Required	Default	I/O
apQuery:WorkflowActionForRuleSet	The workflow action query criteria.	Yes	Not applicable	Input
apData:ListOfWorkflowActionSet	The extracted workflow action data.	Not applicable	Not applicable	Output

Return Value of the Call

The workflow action sets are returned.

WorkflowActionSetUpsert

Updates existing workflow action configuration data or inserts new data.

Usage

Use the WorkflowActionSetUpsert method to update or insert workflow action configuration data. The Manage Workflow Rules privilege is required.

The workflow action set is specified by the apData:WorkflowActionSet argument, and the associated workflow rule is specified with the <WorkflowName> child element of apData:WorkflowActionsSet.

The workflow action set is specified by the <ListOfWorkflowActions> element, which has a <WorkflowAction> element for each action. Each action is specified by the following elements and a set of elements depending on the type of action:

- **Name.** The name of the action.
- **Active.** Whether the action is active.

The following child elements of <WorkflowAction> specify the data for the various types of action:

- **WorkflowCreateTaskAction.** The Create Task action:
 - **Owner.** The owner for the task (required).
 - **OwnerType.** The type of owner (required). Valid values are:
 - **R.** Corresponds to the Relative User on Record value in the UI.
 - **U.** Corresponds to the Specific User value in the UI.
 - **Type.** The activity type.
 - **DueDate.** The due date for the task (required).

- **Priority.** The activity priority (required): 1 - High, 2 - Medium, 3 - Low.
- **Status.** The activity status.
- **TaskCurrency.** The three-character currency code.
- **Private.** Whether the task is visible only to the task owner.
- **Subject.** The subject line of the task (required).
- **Description.** The description of the task.
- **WorkflowAssignBookAction.** The Assign a Book action:
 - **BookName.** The name of the book.
 - **AssignmentOption.** The assignment option (required): Add, Replace Type, Replace All, Replace Book, Remove, or Remove All.
 - **ReplaceBookName.** The name of the book to be replaced, if Replace Book is specified for <AssignmentOption>.
 - **ApplyTo.** How the assignment option is to be applied (required): Manual Associations, Automatic Associations, or Both.
- **WorkflowSyncOwnerPartnerAction.** The Owner Partner Account Sync action.
- **WorkflowSyncBookPartnerAction.** The Book Partner Sync action.
- **WorkflowSyncPartnerBooksAction.** The Partner Book Sync action.
- **WorkflowCreateIntegrationEventAction.** The Create Integration Event action:
 - **ListOfQueues.** The queues to which the integration events are written.
 - **QueueName.** The queue name (required).
 - **ListOfFields.** The fields tracked by the Create Integration Event action.
 - **FieldName.** The system name of the field (required).
 - **AlwaysInclude.** Whether the field is included in the integration event, even if the value of the field has not been changed.
 - **TrackChanges.** Whether an integration event is generated each time the field is updated.
- **WorkflowFieldUpdateAction.** The Update Values action:
 - **FieldName.** The system name of the field (required).
 - **Value.** The new value for the field.
 - **OverwriteExistingValues.** Whether the existing value in the field is to be overwritten with the new value.
- **WorkflowWaitAction.** The Wait action:
 - **ReevaluateRuleConditionsAfterWait.** Whether Oracle CRM On Demand reevaluates the rule conditions after the wait period ends.
 - **WaitType.** The duration of wait (required). Valid values are: Period, Period Expression, Date Time, or Date Time Expression.

- **DateTime.** The date and time if <WaitType> specifies Date Time.
- **WaitExpression.** An expression if <WaitType> specifies Period Expression or Date Time Expression. When the <WaitType> value is Period, the <WaitExpression> element indicates a period based on the value PYMDTHM. For example:
 - ❑ PYM10DTHM indicates 10 days.
 - ❑ P1Y3MDTHM indicates 1 year and 3 months.
 - ❑ PYMDT5H15M indicates 5 hours and 15 minutes.
- **WorkflowSendEmailAction.** The Send Email action:
 - **FromType.** The sender email address (required). Valid values are:
 - ❑ **FROM_EMAIL_ADDRESS.** Corresponds to the Current User value in the UI.
 - ❑ **FROM_DEFAULT.** Corresponds to the Default Email Address value in the UI.
 - ❑ **SPECIFIC_EMAIL_ADDRESS.** Corresponds to the Specific Email Address value in the UI.
 - **From.** The email address, if Specific Email Address is specified for <FromType>.
 - **ToType.** The recipient of the email (required). Valid values are:
 - ❑ **R.** Corresponds to the Relative User on Record value in the UI.
 - ❑ **U.** Corresponds to the Specific User value in the UI.
 - ❑ **E.** Corresponds to the Specific Email Address value in the UI.
 - **To.** The email address, if Specific Email Address is specified for <ToType>.
 - **Subject.** The subject of the email (required).
 - **MessageBody.** The message body (required).

For more information about workflow rules and actions, see *Oracle CRM On Demand Online Help*.

The following workflow actions are available only in Oracle CRM On Demand for Partner Relationship Management:

- Partner Book Sync
- Book Partner Sync
- Owner Partner Account Sync

For more information on these actions and how you can use them, see *Oracle CRM On Demand for Partner Relationship Management Configuration Guide*, available from the Oracle CRM On Demand documentation library on Oracle Technology Network.

Arguments

[Table 126](#) describes the arguments taken by the WorkflowActionSetUpsert method.

Table 126. Arguments Taken by the WorkflowActionSetUpsert Method

Name	Description	Required	Default	I/O
apData:WorkflowActionSet	The workflow action data to be updated or inserted.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

WorkflowRuleSetDelete

Deletes workflow rule configuration data.

Usage

Use the WorkflowRuleSetDelete method to delete workflow rules and the associated workflow action configuration data. The Manage Workflow Rules privilege is required.

The workflow rule set is specified with the <RecordType>, <TriggerEvent>, and <ParentRecordType> child elements of apQuery:WorkflowRuleSet. <RecordType> and <TriggerEvent> are required and <ParentRecordType> is optional. You can delete a specific workflow rule by specifying a value for <WorkflowName>.

See [“WorkflowRuleSetUpsert” on page 278](#) for information about the workflow rule data specified in apQuery:WorkflowRuleSet.

Arguments

[Table 127](#) describes the arguments taken by the WorkflowRuleSetDelete method.

Table 127. Arguments Taken by the WorkflowRuleSetDelete Method

Name	Description	Required	Default	I/O
apQuery:WorkflowRuleSet	The workflow rule set.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

WorkflowRuleSetRead

Extracts workflow rule configuration data.

Usage

Use the WorkflowRuleSetRead method to extract workflow rule configuration data.

The workflow rule set is specified with the <RecordType>, <TriggerEvent> and <ParentRecordType> child elements of apQuery:WorkflowRuleSet. <RecordType> and <TriggerEvent> are required and <ParentRecordType> is optional.

See [“WorkflowRuleSetUpsert” on page 278](#) for information about the workflow rule data specified in apData:WorkflowRuleSet.

Arguments

[Table 128](#) describes the arguments taken by the WorkflowRuleSetRead method.

Table 128. Arguments Taken by the WorkflowRuleSetRead Method

Name	Description	Required	Default	I/O
apQuery:WorkflowRuleSet	The workflow rule query criteria.	Yes	Not applicable	Input
apData:ListOfWorkflowRuleSet	The extracted workflow rule data.	Not applicable	Not applicable	Output

Return Value of the Call

The workflow rules data is returned.

WorkflowRuleSetReadAll

Extracts all of the workflow rule configuration data for a company.

Usage

Use the WorkflowRuleSetReadAll method to read all of the workflow rule sets.

There is no input argument; the method simply returns all of the workflow rule data for all record types.

See [“WorkflowRuleSetUpsert” on page 278](#) for information about the workflow rule data specified in `apData:WorkflowRuleSet`.

Arguments

[Table 129](#) describes the arguments taken by the `WorkflowRuleSetReadAll` method.

Table 129. Arguments Taken by the `WorkflowRuleSetReadAll` Method

Name	Description	Required	Default	I/O
<code>apData:ListOfWorkflowRuleSet</code>	The extracted data for all workflow rules for all objects.	Not applicable	Not applicable	Output

Return Value of the Call

The workflow rule sets are returned.

WorkflowRuleSetUpsert

Updates existing workflow rule configuration data or inserts new data. The Manage Workflow Rules privilege is required. Additionally, the Workflow Cancel Save privilege is required if the Cancel save setting is to be updated or inserted.

Usage

Use the `WorkflowRuleSetUpsert` method to update or insert workflow rule configuration data. The Manage Workflow Rules privilege is required.

The `apData:WorkflowRuleSet` argument has the following child elements:

- **RecordType**. The record type (required).
- **TriggerEvent**. The event that triggers the workflow rule (required). The possible values and the associated trigger events are as follows:

Trigger Event	TriggerEvent LIC
When New Record Saved	WriteRecordNew
When Modified Record Saved	WriteRecordUpdated
Before Modified Record Saved	PreWriteRecord
Before Record Is Deleted	PreDeleteRecord

Trigger Event	TriggerEvent LIC
After Association With Parent	Associate
After Dissociation From Parent	Dissociate
When Record Is restored	RestoreRecord
When Records are merged	MergeRecords

- **ParentRecordType.** The parent record type. This is specified if the trigger event is After Association With Parent or After Dissociation From Parent. These trigger events are supported only for the Account, Contact, and Opportunity record types.
- **ListOfWorkflowRules.** The list of workflow rules for the record type and trigger event. Each <WorkflowRule> element has the following child elements:
 - **WorkflowName.** The name of the workflow rule (required). This name is used to reference an associated workflow action set.
 - **Active.** Whether the workflow rule is enabled or disabled.
 - **Exit.** Whether exiting a workflow in a sequence of workflows is enabled. If the condition on the rule is met, the active actions on the workflow rule are performed. However, none of the subsequent workflow rules that are based on the same record type and the same trigger event, and where applicable, the same parent record type, are triggered.
 - **WorkflowRuleCondition.** The condition for the workflow rule.
 - **CancelSave.** Whether cancel save processing is performed, if the condition is not met.
 - **DisplayMessage.** The message returned if the condition is met, and if cancel save processing is performed.

NOTE: The order of invocation of workflow rules is determined by the sequence of <WorkflowRule> elements.

New workflow rules are inserted. Existing workflows rules are updated, but existing workflow rules that are not in the request are deleted.

For more information about workflow rules, see *Oracle CRM On Demand Online Help*.

Arguments

Table 130 describes the arguments taken by the WorkflowRuleSetUpsert method.

Table 130. Arguments Taken by the WorkflowRuleSetUpsert Method

Name	Description	Required	Default	I/O
apData:WorkflowRuleSet	The workflow rule data to be updated or inserted.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

Return Value of the Call

The status is returned.

Data Loader API Calls

In Oracle CRM On Demand you can use the Oracle Data Loader On Demand tool to perform bulk data import operations. Client applications can also take advantage of the API used by the tool to make Web services calls to submit bulk import requests.

Users with the Admin Import privilege can download the Oracle Data Loader On Demand tool from the Oracle CRM On Demand UI. When you download the tool, a WSDL file is provided as part of the.zip file: OracleDataLoaderOnDemandImportServices.wsdl.

Table 131 lists the calls available with the Data Loader API.

Table 131. Calls Available in the Data Loader API

Name	Description
BulkOpImportCreateRequest	Creates a bulk import request.
BulkOpImportSendData	Imports the data. The BulkOpImportCreateRequest call must be made before BulkOpImportSendData. Subsequently, BulkOpImportSendData calls can be made repeatedly to send data for importing.
BulkOpImportGetRequestDetail	Gets the details of a bulk import request.

BulkOpImportCreateRequest

Creates a bulk import request.

Usage

Use the `BulkOpImportCreateRequest` method to create a bulk import request.

The arguments taken by the method correspond to elements in the WSDL file, as described in [Table 132](#).

The `xsdLocal1:ImportOptions` argument specifies the import options, which are similar to the options available in the Import Wizard in the Oracle CRM On Demand application:

- **CSVDelimiter**. The type of CSV delimiter used in the file: comma (,) or semi-colon (;).
- **DateTimeFormat**. The date and time format used in the CSV file: usa, can, eur, uk, other.
- **DuplicateCheckOption**. Whether the Row Id (rowid) or external ID (externalid) are used for duplicate checking.
- **ErrorLogLevel**. The level of error logging:
 - all. All messages
 - errors. Error messages only.
 - errorswarnings. Error messages and warning messages only.

Typical customer import data is contained in a CSV file with the following format:

```
Col umnName1, Col umnName2, ...
```

```
Data1, Data2, ...
```

```
Data1, Data2, ...
```

```
...
```

The `xsdLocal1:ListOfFieldMappings` element specifies the mapping of CSV column names to field names in Oracle CRM On Demand. For example:

```
<ns1: Li stOfFi el dMappi ngs>
```

```
  <ns1: Fi el dMappi ng>
```

```
    <ns1: CSVCol umnName>
```

```
      Web Si te
```

```
    </ns1: CSVCol umnName>
```

```
  <ns1: Fi el dName>
```

```
    Home Page
```

```
  </ns1: Fi el dName>
```

```
</ns1: Fi el dMappi ng>
```

```
</ns1: Li stOfFi el dMappi ngs>
```

`CSVColumnHeaders` specifies the list of CSV column names, for example:

```
<ns0: CSVCol umnHeaders>
```

"Account Name", "External Unique Id", "Web Site", "Account Location"

</ns0:CSVColumnHeaders>

The mappings are validated and if everything is correct, the import request is created.

Arguments

Table 132 describes the arguments taken by the BulkOpImportCreateRequest method.

Table 132. Arguments Taken by the BulkOpImportCreateRequest Method

Name	Description	Required	Default	I/O
DataFileName	The name of the data file to be imported, for example, account1.csv	Yes	Not applicable	Input
RecordType	The record type to be imported.	Yes	Not applicable	Input
xsdLocal1:ImportOptions	The list of import options.	No	Not applicable	Input
xsdLocal1:ListOfFieldMappings	The mapping of CSV file column names and their corresponding fields.	Yes	Not applicable	Input
CSVColumnHeaders	The list of column names from the CSV file.	Yes	Not applicable	Input
EnableImportAudit	Enables or disables auditing of record creation/update during import requests. This setting takes effect only if the user has the privilege Manage Record Auditing for Imports, otherwise it is ignored. Accepted values are true or false.	No	true	Input
CSVRowCount	The number of data records in the data file to be imported.	Yes	Not applicable	Input
Operation	The operation for the current request, such as insert, update.	Yes	Not applicable	Input
ClientVersion	The current Data Loader client version number.	No	Not applicable	Input
RequestId	The ID of the request created.	Not applicable	Not applicable	Output

Table 132. Arguments Taken by the BulkOpImportCreateRequest Method

Name	Description	Required	Default	I/O
Status	The status of this Web service request.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service request, or a message indicating success.	Not applicable	Not applicable	Output

Return Value of the Call

The request ID, status, and any error message are returned.

BulkOpImportSendData

Imports the data for a bulk import request.

Usage

Use the BulkOpImportSendData method to import the data for a bulk import request.

The BulkOpImportCreateRequest call must be invoked before BulkOpImportSendData by the client.

...

The ListOfCSVData element specifies the row data from the CSV file. For example:

```
<ns1:ListOfCSVData>
  <ns1:CSVData>
    <ns1:Row>
      "High Quality Moving Company", "HQMC", "www.highmoving.net", "San Jose"
    </ns1:Row>
  </ns1:CSVData>
</ns1:ListOfCSVData>
```

The CSV data is processed and the actual import performed.

Arguments

Table 133 describes the arguments taken by the BulkOpImportSendData method.

Table 133. Arguments Taken by the BulkOpImportSendData Method

Name	Description	Required	Default	I/O
RequestId	The request ID that was created.	Yes	Not applicable	Input/Output
TotalNumberOfCSVDataLists	The total number of BulkOpImportSendData calls needed to send all CSV file data.	Yes	Not applicable	Input
xsdLocal1:ListOfCSVData	The actual list of CSV data contained in the CSV file.	Yes	Not applicable	Input
StartRecordNumber	The starting line number of the CSV data.	Yes	Not applicable	Input
ClientVersion	The current data loader client version number.	No	Not applicable	Input
Status	The status of this Web service call.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service call, or a message indicating success.	Not applicable	Not applicable	Output

Return Value of the Call

The request ID, status, and any error message are returned.

BulkOpImportGetRequestDetail

Gets the details of a bulk import request.

Usage

Use the BulkOpImportGetRequestDetail method to get the details of a bulk import request, which is identified by the RequestId argument.

Arguments

Table 134 describes the arguments taken by the BulkOpImportGetRequestDetail method.

Table 134. Arguments Taken by the BulkOpImportGetRequestDetail Method

Name	Description	Required	Default	I/O
RequestId	The request ID.	Yes	Not applicable	Input/Output
ClientVersion	The current data loader client version number.	No	Not applicable	Input
RequestStatus	The status of the request.	Not applicable	Not applicable	Output
RecordType	The record type of the request.	Not applicable	Not applicable	Output
RequestedBy	The requestor of the request.	Not applicable	Not applicable	Output
CompletionTime	The time and date the request completed.	Not applicable	Not applicable	Output
RequestTime	The time and date the request was submitted to be processed.	Not applicable	Not applicable	Output
NumberSubmitted	The number of records to be imported by the request.	Not applicable	Not applicable	Output
NumberProcessed	The number of records processed by the request.	Not applicable	Not applicable	Output
NumberPartialImported	The number of records partially imported by the request.	Not applicable	Not applicable	Output
NumberNotImported	The number of records not imported by the request.	Not applicable	Not applicable	Output
NumberSuccessImported	The number of records successfully imported by the request.	Not applicable	Not applicable	Output
Status	The status of this Web service call.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service call.	Not applicable	Not applicable	Output

Return Value of the Call

The output arguments.

Oracle CRM On Demand Objects Accessible Through Web Services

This chapter contains reference information about the objects accessible through the Web Services On Demand API. These objects correspond to record types and enable access to data stored within an instance of Oracle CRM On Demand. This chapter contains the following topics:

- [Reference Information About the Parent Objects on page 287](#)
- [Core Parent Objects \(Web Services v1.0\) on page 290](#)
- [Life Sciences Edition Parent Objects \(Web Services v1.0\) on page 405](#)
- [Financial Services Edition Parent Objects \(Web Services v1.0\) on page 408](#)
- [Automotive Edition Parent Objects \(Web Services v1.0\) on page 418](#)
- [Child Objects \(Web Services v1.0\) on page 424](#)
- [Core Parent Objects \(Web Services v2.0\) on page 431](#)
- [Partner Relationship Management Edition Objects \(Web Services v2.0\) on page 475](#)
- [Life Sciences Edition Parent Objects \(Web Services v2.0\) on page 509](#)
- [Financial Services Edition Parent Objects \(Web Services v2.0\) on page 557](#)
- [Automotive Edition Parent Objects \(Web Services v2.0\) on page 584](#)
- [Child Objects \(Web Services v2.0\) on page 588](#)

Reference Information About the Parent Objects

The reference information about the parent objects (starting with [“Account” on page 291](#)) includes:

- A description of each object, as well as information on usage of the object.
- Information about the relationships between objects; for each object, the associated parent and child objects are listed
- The methods that can be invoked to insert, update, delete, and find data. For more information on these methods, see [Chapter 6, “Web Services On Demand API Calls.”](#)
- The fields that are accessible for the objects:
 - The required and read-only fields
 - The user keys, see [“Oracle CRM On Demand User Keys” on page 288](#)
 - The audit fields, see [“Audit Fields” on page 288](#)
 - The status key, see [“Oracle CRM On Demand Status Keys” on page 289](#)
 - The pick map fields, see [“Oracle CRM On Demand Pick Maps” on page 290](#)

- For Web Services v1.0 only, the filterable fields, see [“Filterable Fields” on page 290](#)
- 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 *ObjectId*; 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:

- *ObjectId*
- *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 *ObjectId* or *ExternalSystemId* of a record might not be known, but the values for some other user key might be known, in which case the record can be successfully queried or updated using that user key. For example, for a Note child object, the Subject and Description fields form a user key, because they can be used in conjunction with each other to determine whether the record is unique or not. Such a combination is not guaranteed to provide complete uniqueness, but it can be used to query for uniqueness.

The user keys for each object are detailed for each object in [“Core Parent Objects \(Web Services v1.0\)” on page 290](#), and [“Core Parent Objects \(Web Services v2.0\)” on page 431](#).

Audit Fields

The audit fields for an object provide information about who created an instance of the object, when it was created, who has last updated an instance of the object, and when it was last updated. All objects, both parent and child level, accessible through the Web services API contain the read-only audit fields contained in [Table 135](#).

Table 135. 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: <i>“Creator Full Name, CreatedDate”</i>
CreatedById	The Row ID of the user who created the record.

Table 135. Audit Fields for the Oracle CRM On Demand Objects

Field Name	Description
CreatedDate	The DateTime stamp of when the record was created.
ModifiedBy	This field is a combination of the full name of the person who modified this instance of the object, and the date on which the instance was modified. This information is contained within the field in the following format: <i>"Modified By Full Name, ModifiedDate"</i>
ModifiedById	The Row ID of the user who last modified the record.
ModifiedDate	The DateTime stamp of when the record was last modified.

Oracle CRM On Demand Status Keys

An Oracle CRM On Demand *status key* is a field or a number of fields that is returned following an operation on an Oracle CRM On Demand object.

For Web Services v1.0, the status key of objects contained through the Web services API contains all user key and audit fields in addition to some other fields that are identified as status keys for the object. The status keys for the Oracle CRM On Demand objects are outlined in ["Core Parent Objects \(Web Services v1.0\)" on page 290](#).

For Web Services v2.0, a standard set of status key fields is returned for all parent and child objects:

- Id
- CreatedBy
- CreatedById
- CreatedDate
- ModifiedBy
- ModifiedById
- ModifiedDate
- ModId

This set of fields is standard across all Web Services v2.0 objects as long as the field is available in the associated WSDL file.

The ModId field is the modification key, which can be used with all Web Services v2.0 API methods apart from QueryPage. If this key is specified in the request, it helps check and protect against concurrent updates.

Oracle CRM On Demand Pick Maps

An Oracle CRM On Demand *pick map* allows you to set a foreign key for an object using a different field from the foreign key field.

For example, when updating an account, you might want to set the owner of the account to a specific user. If the UserId of the user is known it can be set in the OwnerId field, which is the foreign key. However, if the UserId is not known, and only the alias of the user is known, that alias can be entered in the Owner field, which is a pick map field. When Oracle CRM On Demand recognizes that the Owner field has been set, it automatically sets the OwnerId field to the UserId for the user.

Pick maps can be used by a number of Oracle CRM On Demand objects to update foreign key references in this way. For each object, a list of pick map fields, and the foreign key fields that they map to, are detailed in [“Core Parent Objects \(Web Services v1.0\)” on page 290](#) and [“Core Parent Objects \(Web Services v2.0\)” on page 431](#).

Filterable Fields

A filterable field is a field in which you can apply a search query. For the Web Services V1.0 API, all fields in parent objects are filterable and some fields on child objects are filterable; these fields are shown in the tables of filterable fields for each object in [“Core Parent Objects \(Web Services v1.0\)” on page 290](#). For the Web Services V2.0 API, all fields in parent objects are filterable.

Core Parent Objects (Web Services v1.0)

This topic includes the standard objects of Oracle CRM On Demand. Objects that are only available with industry-specific editions of the application are covered in the following topics:

- [“Life Sciences Edition Parent Objects \(Web Services v1.0\)” on page 405](#)
- [“Financial Services Edition Parent Objects \(Web Services v1.0\)” on page 408](#)
- [“Automotive Edition Parent Objects \(Web Services v1.0\)” on page 418](#)

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

- [“Account” on page 291](#)
- [“Activity” on page 310](#)
- [“Asset” on page 319](#)
- [“Book” on page 322](#)
- [“Campaign” on page 325](#)
- [“Contact” on page 334](#)
- [“Current User” on page 353](#)
- [“CustomObject1 - CustomObject3” on page 355](#)
- [“Lead” on page 364](#)

- ["Note" on page 371](#)
- ["Opportunity" on page 372](#)
- ["Product" on page 385](#)
- ["Product Category" on page 388](#)
- ["Service Request" on page 390](#)
- ["Solution" on page 394](#)
- ["Territory" on page 398](#)
- ["User" on page 399](#)
- ["User Group" on page 402](#)

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.

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Parent Objects

Account, [CustomObject1](#) - [CustomObject3](#)

Child Components

[Activity](#), [Address](#), [Asset](#), [Attachment](#), [Book](#), [Competitor](#), [Contact](#), [CustomObject3](#), [Lead](#), [Multiple Contact Roles](#), [Note](#), [Opportunity](#), [Team](#), [Related Account](#), [Revenue](#), [Service Request](#), and [Team](#).

Methods Called

Table 136 details the methods called by the Account service.

Table 136. Methods Called by Account Service

Method	Name as Defined in Service
"Delete" on page 85	AccountDelete
"DeleteChild" on page 88	AccountDeleteChild
"Insert" on page 89	AccountInsert
"InsertChild" on page 90	AccountInsertChild
"InsertOrUpdate" on page 90	AccountInsertOrUpdate
"QueryPage" on page 91	AccountQueryPage
"Update" on page 101	AccountUpdate
"UpdateChild" on page 102	AccountUpdateChild

Fields

Table 137 details the required and read-only fields for the account object.

Table 137. Required and Read-Only Fields for the Account Object

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

Table 137. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	DealerName	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	AccountId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 137. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Contact	Age	Read-only
	OwnerFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	LastActivityDate	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
CustomObject3	AccountCustomObject3CreatedByld	Read-only
	AccountCustomObject3CreatedDate	Read-only
	AccountCustomObject3ModifiedByld	Read-only
	AccountCustomObject3ModifiedDate	Read-only
	CustomObject3ld	Read-only
Lead	OwnerFullName	Read-only
	SalesRepFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	ReferredByFullName	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
	CompanyNameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
Multiple Contact Roles	ContactRole	Required
	Contactld	Required
	Audit Fields	Read-only
Opportunity	OwnerFullName	Read-only

Table 137. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
	PrimaryRevenueExpectedValue	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	FuriganaAccountName	Read-only
	NameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
RelatedAccount	AccountRelationshipId	Read-only
	RelatedAccountId	Read-only
Revenue	RevenueId	Required
	PartNumber	Required
	Revenue	Required
	ContactFullName	Required
	Audit Fields	Read-only
ServiceRequest	Name	Read-only
	OwnerFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	LastAssessmentDate	Read-only
	AssignmentStatus	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	SRNumberShadow	Read-only
	AbstractShadow	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only

Table 138 details the status key for the account object, and the child component on which this key resides.

Table 138. Status Key for the Account Object

Child Component	Field Name
Account (parent)	Audit Fields
	ExternalSystemId
	AccountId
	IntegrationId
	LastUpdated
AccountNote	Audit Fields
	ExternalSystemId
	AccountNoteId
	IntegrationId
AccountTeam	Audit Fields
	AccountTeamId
Activity	Audit Fields
	ExternalSystemId
	ActivityId
	IntegrationId
Asset	Audit Fields
	ExternalSystemId
	AssetId
	IntegrationId
Attachment	Audit Fields
	Id
	AccountId
	ModId
Book	Audit Fields
	BookId
	ModId

Table 138. Status Key for the Account Object

Child Component	Field Name
Competitor	Audit Fields
	AccountCompetitorId
	CompetitorExternalId
	CompetitorId
	CompetitorIntegrationId
Contact	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
CustomObject3	AccountCustomObject3CreatedById
	AccountCustomObject3CreatedDate
	AccountCustomObject3ModifiedById
	AccountCustomObject3ModifiedDate
Lead	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
Opportunity	Audit Fields
	AccountId
	ExternalSystemId
	IntegrationId
	OpportunityId
RelatedAccount	Audit Fields
	AccountRelationshipId

Table 138. Status Key for the Account Object

Child Component	Field Name
Revenue	Audit Fields
	ExternalId
	IntegrationId
	RevenueId
ServiceRequest	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	ServiceRequestId
Partner	Audit Fields
	AccountPartnerId
	PartnerExternalId
	PartnerId
	PartnerIntegrationId
	Updated

[Table 139](#) details the pick map fields for the account object and the child objects on which they reside.

Table 139. Pick Map Fields for the Account Object

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

Table 139. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	DelegatedBy	DelegatedById
	FundRequest	FundRequestId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	PrimaryContactIntegrationId	PrimaryContactId
	PrimaryContactExternalId	PrimaryContactId
	LeadIntegrationId	LeadId
	LeadExternalId	LeadId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SRIntegrationId	SRId
	SRExternalId	SRId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Asset	AccountIntegrationId	AccountId
	Manufacturer	ManufacturerId
	PreferredServiceDealer	PreferredServiceDealerID
Book	BookName	BookId
Competitor	RelatedAccountExternalId	RelatedAccountId
	RelatedAccountSystemId	RelatedAccountId

Table 139. Pick Map Fields for the Account Object

Child Component	Pick Map Field	Maps To
Contact	AccountExternalId	AccountId
	AccountIntegrationId	AccountId
	AccountName	AccountId
	ManagerExternalId	ManagerId
	ManagerIntegrationId	ManagerId
	OwnerEmailAddress	OwnerId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SourceCampaignName	SourceCampaignId
	SourceCampaignExternalId	SourceCampaignId
	TimeZoneName	TimeZoneId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
CustomObject3	Owner	OwnerId

Table 139. 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 139. 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 139. 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 140 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 140. Filterable Fields and User Key Fields on the Account Object's Child Components

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

Table 140. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity	CallType	IntegrationId
		ActivityId
		ExternalSystemId
Address	AddressId	AddressId
	ExternalId	ExternalSystemId
	IntegrationId	IntegrationId
	City	
	Country	
	ModifiedDate	
	Province	
	StateProvince	
	ZipCode	
Asset	AssetId	AssetId
	Contract	IntegrationId
	Date	ExternalSystemId
	ModifiedDate	
	PartNumber	
	ProductCategory	
	Product	
	ProjectManager	
	PurchaseDate	
	Price	
	Quantity	
	SalesRep	
	SerialNumber	
	ShipDate	
	Status	
	Type	
	Warranty	

Table 140. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Contact	AccountContactModifiedById	ExternalSystemId
	AccountContactModifiedDate	IntegrationId
	ContactType	
	ContactFirstName	
	ContactLastName	
	JobTitle	
	Owner	
Competitor	ModifiedDate	CompetitorId
		IntegrationId
		CompetitorExternalSystemId
CustomObject3	AccountCustomObject3ModifiedById	CustomObject3Id
	AccountCustomObject3ModifiedDate	ExternalSystemId
	CustomObject3Id	IntegrationId
	CustomObject3Name	
	ExternalSystemId	
	IntegrationId	
	Type	

Table 140. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	Campaign	Id
	EstimatedCloseDate	LeadId
	Rating	IntegrationId
	Source	ExternalSystemId
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
	Id	
Multiple Contact Roles	ContactId	ContactId
	ContactExternalId	ContactIntegrationId
	ContactIntegrationId	ContactExternalId
	ContactRole	
	ModifiedDate	
Opportunity	Owner	OpportunityId
	Revenue	Id
	CloseDate	IntegrationId
	Forecast	ExternalSystemId
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	Status	
	OpportunityId	
	Id	

Table 140. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Partner	AccountPartnerId	PartnerIntegrationId
	PrimaryContactId	PartnerExternalSystemId
	EndDate	
	PartnerId	
	PartnerExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	
Related Account	AccountRelationshipId	AccountRelationshipId
	Comments	RelatedAccountId
	EndDate	RelatedAccountExternalId
	ModifiedDate	RelatedAccountIntegrationId
	RelatedAccountExternalId	
	RelatedAccountId	
	RelatedAccountIntegrationId	
	RelationshipRole	
	RelationshipStatus	
	RelationshipType	
	ReverseRelationshipRole	
	StartDate	

Table 140. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Revenue	ContactFullName	RevenueId
	Description	ProductId
	ExternalId	ProductExternalId
	IntegrationId	ProductIntegrationId
	Forecast	IntegrationId
	Frequency	ExternalSystemId
	ModifiedDate	
	NumberOfPeriods	
	Product	
	ProductCategoryId	
	ProductCategoryExternalId	
	ProductCategoryIntegrationId	
	ProductExternalId	
	ProductId	
	ProductIntegrationId	
	PurchasePrice	
	Quantity	
	Revenue	
	RevenueId	
	StartCloseDate	
	Status	
	Type	

Table 140. Filterable Fields and User Key Fields on the Account Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Service Request	Subject	Id
	Area	ServiceRequestId
	Owner	IntegrationId
	Priority	ExternalSystemId
	Type	
	Cause	
	Source	
	Status	
	Id	
	ServiceRequestId	

Table 141 details the picklists available for the account object.

Table 141. Picklists Available for the Account Object

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

Table 141. Picklists Available for the Account Object

Child Component	Field Name
Revenue	Type
	Status
	Frequency

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

Related Topic

[Contact](#)

Activity

The activity object stores information on an activity that a user must carry out, for example, a call-back activity for an account. When an activity is created, the user must set the Activity field explicitly to Task or Appointment.

Usage

Oracle On Demand Web Services uses activities to organize, track, and resolve a variety of tasks, from finding and pursuing opportunities to closing service requests. If a task requires multiple steps that one or more people can carry out, activities greatly simplify the job. Activities can help to:

- Define and assign the task
- Provide information to complete the task
- Track the progress of the task
- Track costs and bill for the task

Parent Objects

[Account](#), [Campaign](#), [Contact](#), [Lead](#), [Opportunity](#), and [Service Request](#)

Child Components

[Attachment](#), [Book](#), [Contact](#), [ProductsDetailed](#), [PromotionalItemsDropped](#), [SampleDropped](#), [Solution](#), and [User](#).

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Methods Called

Table 142 details the methods called by the Activity service.

Table 142. Methods Called by Activity Service

Method	Name as Defined in Service
"Delete" on page 85	ActivityDelete
"DeleteChild" on page 88	ActivityDeleteChild
"Insert" on page 89	ActivityInsert
"InsertChild" on page 90	ActivityInsertChild
"InsertOrUpdate" on page 90	ActivityInsertOrUpdate
"QueryPage" on page 91	ActivityQueryPage
"Update" on page 101	ActivityUpdate
"UpdateChild" on page 102	ActivityUpdateChild

Fields

Table 143 details the required and read-only fields for the activity object.

Table 143. Required and Read-Only Fields for the Activity Object

Child Component	Field Name	Type
Activity (parent)	Subject	Required
	Activity	Required
	ActivityId	Read-only
	AddressId	Read-only
	CallType	Read-only
	ContactFirstName	Read-only
	ContactLastName	Read-only
	LeadFirstName	Read-only
	LeadLastName	Read-only
	MedEdEventName	Read-only
	OpportunityName	Read-only
	FundRequest	Read-only
	SmartCall	Read-only
	AssignedQueue	Read-only
	QueueHoldTime	Read-only
	QueueStartTime	Read-only
	TotalHoldTime	Read-only
	ResolutionCode	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	ActivityId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 143. 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
PromotionalItemDropped	ProductName	Required
	Quantity	Required
	ProductAllocationId	Required
	Audit Fields	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 144 details the status key for the activity object.

Table 144. Status Key for the Activity Object

Child Component	Field Name
Activity (parent)	Audit Fields
	ActivityId
	ExternalSystemId
	IntegrationId
Attachment	Audit Fields
	Id
	ActivityId
Book	Audit Fields
	BookId
	ModId
Contact	Audit Fields
	ActivityContactId
	ContactIntegrationId
	ContactExternalSystemId
	ContactId
User	Audit Fields
	UserId
	UserExternalSystemId
	UserIntegrationId
ProductDetailed	Audit Fields
	ProductDetailedId
	ExternalId
SampleDropped	Audit Fields
	SampleDroppedId
	ExternalId
Solution	Audit Fields
	SolutionId
	ExternalId

Table 145 details the pick map fields for the activity object and the child objects on which they reside.

Table 145. Pick Map Fields for the Activity Object

Child Component	Pick Map Field	Maps To
Activity (parent)	AccountName	AccountId
	AccountLocation	AccountId
	AccountExternalSystemId	AccountId
	AccountIntegration	AccountId
	Owner	OwnerId
	CampaignExternalSystemId	CampaignId
	CampaignIntegrationId	CampaignId
	CampaignName	CampaignId
	LeadExternalSystemId	LeadId
	LeadIntegrationId	LeadId
	MedEdEventExternalSystemId	MedEdEventId
	MedEdEventIntegrationId	MedEdEventId
	OpportunityExternalSystemId	OpportunityId
	OpportunityIntegrationId	OpportunityId
	PortfolioExternalSystemId	PortfolioId
	PortfolioIntegrationId	PortfolioId
	ServiceRequestNumber	ServiceRequestId
	ServiceRequestExternalSystemId	ServiceRequestId
	ServiceRequestIntegrationId	ServiceRequestId
	FundRequestExternalSystemId	FundRequestId
	FundRequestIntegrationId	FundRequestId
Book	BookName	BookId
Contact	ContactExternalId	ContactId
	ContactIntegrationId	ContactId
User	UserExternalSystemId	UserId
	UserIntegrationId	UserId
SampleDropped	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId

Table 145. Pick Map Fields for the Activity Object

Child Component	Pick Map Field	Maps To
ProductDetailed	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId
PromotionalItemDropped	ProductName	ProductId
	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId
Solution	ProductIntegrationId	ProductId
	ProductExternalSystemId	ProductId

Table 146 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 146. Filterable Fields and User Key Fields on the Activity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity (parent)	All	ActivityId
		IntegrationId
		ExternalSystemId
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 146. Filterable Fields and User Key Fields on the Activity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact	ContactId	ContactId
	ContactExternalSystemId	ContactExternalSystemId
	ContactIntegrationId	ContactIntegrationId
	ContactFirstName	
	ContactLastName	
	ContactAccountId	
	ContactAccountName	
	ContactAccountLocation	
	ContactAccountIntegrationId	
	ContactAccountExternalSystemId	
	ModifiedDate	
ProductDetailed	ProductDetailedId	ProductDetailedId
	ProductDetailedExternalSystemId	ProductDetailedExternalSystemId
	ProductId	ProductId
	ProductExternalSystemId	ProductExternalSystemId
	ModifiedDate	Name
		Indication
SampleDropped	SampleDroppedId	SampleDroppedId
	SampleDroppedExternalSystemId	SampleDroppedExternalSystemId
	ProductId	ProductId
	ProductExternalSystemId	ProductExternalSystemId
	ModifiedDate	Quantity
		ProductName

Table 146. Filterable Fields and User Key Fields on the Activity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Solution	SolutionId	SolutionId
	SolutionExternalSystemId	SolutionExternalSystemId
	ModifiedDate	
	ProductLine	
	PrimaryProductName	
	Name	
	ProductId	
	ProductIntegrationId	
	ProductExternalId	
User	ModifiedDate	UserId
	UserId	UserExternalSystemId
	UserExternalSystemId	UserIntegrationId
	UserIntegrationId	
	UserEmail	
	UserFirstName	
	UserLastName	
	UserRole	

Table 147 details the picklists available for the activity object.

Table 147. Picklists Available for the Activity Object

Field Name
AccountName
AccountLocation
AccountIntegrationId
AccountExternalSystemId
OpportunityName
ServiceRequestNumber
ServiceRequestIntegrationId
ServiceRequestExternalSystemId
DelegatedByExternalSystemId

Table 147. Picklists Available for the Activity Object

Field Name
PrimaryContactIntegrationId
PrimaryContactExternalSystemId
MedEdEventIntegrationId
MedEdEventExternalSystemId
FundRequestExternalId
LeadIntegrationId
LeadExternalSystemId
CampaignIntegrationId
CampaignExternalSystemId
ActivitySubtype
ResolutionCode
PublishInternal
Status
Issue
Indication

For more information on the fields accessible through the Activity Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Asset

The asset object stores information on the assets held by your accounts, for example, the products that an account has purchased. The asset object has no child components.

Usage

Oracle On Demand Web Services uses assets to manage products through their life cycle. It is also used by your accounts to register products, receive product news and literature, track warranty agreements, and receive recommendations on scheduled services.

Parent Objects

[Account](#) and [Contact](#).

Methods Called

Table 148 details the methods called by the Asset service.

Table 148. Methods Called by Asset Service

Method	Name as Defined in Service
"Delete" on page 85	AssetDelete
"Insert" on page 89	AssetInsert
"InsertOrUpdate" on page 90	AssetInsertOrUpdate
"QueryPage" on page 91	AssetQueryPage
"Update" on page 101	AssetUpdate

Fields

Table 149 details the required and read-only fields for the asset object.

Table 149. Required and Read-Only Fields for the Asset Object

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

Table 150 details the status key for the asset object.

Table 150. Status Key for the Asset Object

Child Component	Field Name
Asset (parent)	Audit Fields
	AssetId
	IntegrationId
	ExternalSystemId

Table 151 details the pick map fields for the asset object.

Table 151. Pick Map Fields for the Asset Object

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

Table 152 provides a list of the filterable fields and a list of user key combinations for the asset object.

Table 152. Filterable Fields and User Key Fields on the Asset Object's Child Components

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

Table 153 details the picklists available for the asset object.

Table 153. Picklists Available for the Asset Object

Field Name
Warranty
Contract

For more information on the fields accessible through the Asset Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the asset object.

Book

The book object provides a way of segmenting data according to the organizational units of your business, such as territories or products. Administrators can create book hierarchies based on how they want to organize your information, and then set up users to have the appropriate level of access to books.

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

Usage

To specify when book assignments become active for accounts and contacts, you can update the `StartDate` and `EndDate` fields of the Book child object through Web service calls. If you do not specify a start date for a book assignment, then the assignment becomes active immediately. You can also update the `FuturePrimary` field to specify whether the book that you assign to an account or contact is to become the primary book for that account or contact when the book assignment becomes active.

The `FuturePrimary` field is not visible in the Oracle CRM On Demand UI and the `StartDate` and `EndDate` fields are read-only in the UI. For more information about time-based book assignments, see *Oracle CRM On Demand Online Help*

Parent Objects

[Account](#), [Activity](#), [Contact](#), [Custom Object 1 - 3](#), [Lead](#), [Household](#), [Lead](#), [Opportunity](#), [Product Category](#), [Service Request](#).

Child Components

[BookUser](#), [SubBook](#).

Methods Called

[Table 154](#) details the methods called by the Book service.

Table 154. Methods Called by Book Service

Method	Name as Defined in Service
"Delete" on page 85	BookDelete
"DeleteChild" on page 88	BookDeleteChild
"Insert" on page 89	BookInsert
"InsertChild" on page 90	BookInsertChild
"InsertOrUpdate" on page 90	BookInsertOrUpdate

Table 154. Methods Called by Book Service

Method	Name as Defined in Service
"QueryPage" on page 91	BookQueryPage
"Update" on page 101	BookUpdate
"UpdateChild" on page 102	BookUpdateChild

Fields

[Table 155](#) details the required and read-only fields for the book object.

Table 155. Required and Read-Only Fields for the Book Object

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

[Table 156](#) details the status key for the book object.

Table 156. Status Key for the Book Object

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

Table 157 details the pick map fields for the book object.

Table 157. Pick Map Fields for the Book Object

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

Table 158 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 158. Filterable Fields and User Key Fields on the Book Object's Child Components

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

Table 159 details the picklists available for the book object.

Table 159. Picklists Available for the Book Object

Child Component	Field Name
Book (parent)	BookType
	ParentBookName
BookUser	UserAlias
	BookRole
	AccessProfileName

For more information on the fields accessible through the Book Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the book object.

Campaign

The campaign object provides a mechanism for marketing products and services to customers and prospects. The campaign object is the primary way in which new products and services are marketed to customers and prospects.

Parent Objects

[Contact](#), [Lead](#)

Child Components

[Activity](#), [Attachment](#), [Contact/Recipient](#), [Lead](#), [Note](#), and [Opportunity](#).

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Methods Called

[Table 160](#) details the methods called by the Campaign service.

Table 160. Methods Called by Campaign Service

Method	Name as Defined in Service
"Delete" on page 85	CampaignDelete
"DeleteChild" on page 88	CampaignDeleteChild
"Insert" on page 89	CampaignInsert
"InsertChild" on page 90	CampaignInsertChild
"InsertOrUpdate" on page 90	CampaignInsertOrUpdate
"QueryPage" on page 91	CampaignQueryPage
"Update" on page 101	CampaignUpdate
"UpdateChild" on page 102	CampaignUpdateChild

Fields

Table 161 details the required and read-only fields for the campaign object.

Table 161. 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
	CODIVRTTime	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 161. 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 162 details the status key for the campaign object.

Table 162. Status Key for the Campaign Object

Child Component	Field Name
Campaign (parent)	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
Activity	Audit Fields
	CampaignId
	ExternalSystemId
	Id
	IntegrationId
Attachment	Audit Fields
	Id
	CampaignId
CampaignNote	Audit Fields
	CampaignId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Lead	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Opportunity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 162. Status Key for the Campaign Object

Child Component	Field Name
Recipient	Audit Fields
	CampaignContactId
	ContactExternalId
	ContactIntegrationId
	ContactId

[Table 163](#) details the pick map field for the campaign object.

Table 163. Pick Map Field for the Campaign Object

Child Component	Pick Map Field	Maps To
Campaign (parent)	Owner	OwnerId

Table 163. 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 163. 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 164 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 164. Filterable Fields and User Key Fields on the Campaign Object's Child Components

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

Table 164. Filterable Fields and User Key Fields on the Campaign Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Opportunity	Account	OpportunityName
	Owner	IntegrationId
	Revenue	ExternalSystemId
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
Recipient	ContactId	ContactId
	ModifiedDate	ContactIntegrationId
		ContactExternalSystemId
		CampaignContactId

Table 165 details the picklists available for the campaign object.

Table 165. Picklists Available for the Campaign Object

Child Component	Field Name
Campaign (parent)	CampaignType
	Status
Recipient	DeliveryStatus
	ResponseStatus

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

Related Topic

[Current User](#) and [Opportunity](#)

Contact

The contact object stores information on individuals with whom your organization has a relationship. It allows the user to store information on individuals who are external to your company, but who are associated with the business process. Contacts stored in the Oracle CRM On Demand database can also be associated with an account.

Parent Objects

[Account](#), [Activity](#), [Campaign](#), [CustomObject1 - CustomObject3](#), [Opportunity](#), [Product Category](#), and [Life Sciences Edition Parent Objects \(Web Services v1.0\)](#)

Child Components

[Account](#), [Activity](#), [Address](#), [Asset](#), [Attachment](#), [Book](#), [Campaign](#), [CustomObject3](#), [Interests](#), [Lead](#), [Note](#), [Opportunity](#), [Related Contact](#), [Revenue](#), [Service Request](#) and [Team](#).

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Methods Called

[Table 166](#) details the methods called by the Contact service.

Table 166. Methods Called by Contact Service

Method	Name as Defined in Service
"Delete" on page 85	ContactDelete
"DeleteChild" on page 88	ContactDeleteChild
"Insert" on page 89	ContactInsert
"InsertChild" on page 90	ContactInsertChild
"InsertOrUpdate" on page 90	ContactInsertOrUpdate
"QueryPage" on page 91	ContactQueryPage
"Update" on page 101	ContactUpdate
"UpdateChild" on page 102	ContactUpdateChild

Fields

[Table 167](#) details the required and read-only fields for the contact object.

Table 167. Required and Read-Only Fields for the Contact Object

Child Component	Field Name	Type
Contact (parent)	FirstName	Required
	LastName	Required
	AlternateAddressId	Read-only
	ContactConcatField	Read-only
	ContactFullName	Read-only
	Audit Fields	Read-only
	Manager	Read-only
	PrimaryAddressId	Read-only
Account	AccountId	Read-only
Activity	AccountLocation	Read-only
	Contact	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTIME	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Audit Fields	Read-only
Address	AddressId	Read-only

Table 167. 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 167. 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 167. 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 167. 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 168 details the status key for the contact object.

Table 168. Status Key for the Contact Object

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

Table 168. 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 168. 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 168. Status Key for the Contact Object

Child Component	Field Name
Team	Audit Fields
	ContactTeamId
	UserExternalSystemId
	UserId
	UserIntegrationId

[Table 169](#) details the pick map fields for the contact object.

Table 169. Pick Map Fields for the Contact Object

Child Component	Pick Map Field	Maps To
Contact (parent)	AccountName	AccountId
	Owner	AssignedToAlias
	SourceCampaignName	SourceCampaignId
	ManagerExternalSystemId	ManagerId
Account	AccountExternalId	AccountId
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	AssignedTo	AssignedToId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	DelegatedBy	DelegatedById
	Dealer	DealerId
	FundRequest	FundRequestId
	OpportunityIntegrationId	OpportunityId
	OpportunityExternalId	OpportunityId
	PrimaryContactIntegrationId	PrimaryContactId
	PrimaryContactExternalId	PrimaryContactId
	LeadIntegrationId	LeadId
	LeadExternalId	LeadId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId

Table 169. 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 169. 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 169. 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 170 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 170. Filterable Fields and User Key Fields on the Contact Object's Child Components

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

Table 170. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Account	AccountId	AccountId
	AccountExternalId	AccountExternalId
	Location	Name and Location
	Name	
Activity	CallType	IntegrationId
		ActivityID
		ExternalSystemId
Address	ExternalId	AddressID
	IntegrationId	ExternalSystemId
	City	IntegrationId
	Country	
	ZipCode	
	StateProvince	
	Province	
Asset	AssetId	AssetId
	ExternalSystemId	ExternalSystemId
	IntegrationId	IntegrationId
	ProductId	
	SerialNumber	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Campaign	CampaignContactId	CampaignContactId
	CampaignId	CampaignId
	CampaignExternalSystemId	CampaignExternalSystemId
	CampaignName	CampaignName
	DeliveryStatus	
	ResponseStatus	
	ModifiedDate	

Table 170. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact Note	Subject	Subject and Description
		IntegrationId
		ExternalSystemId
Contact Team	ContactTeamId	UserId
	UserId	UserIntegrationId
	UserExternalSystemId	UserExternalSystemId
	UserIntegrationId	
	UserFirstName	
	UserLastName	
	ContactAccess	
	UserRole	
CustomObject3	ContactCustomObject3ModifiedById	CustomObject3Id
	ContactCustomObject3ModifiedDate	ExternalSystemId
	CustomObject3Id	IntegrationId
	CustomObject3Name	
	ExternalSystemId	
	IntegrationId	
	Type	
Interests	InterestId	ExternalSystemId
	Category	InterestId
	Interests	

Table 170. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	Campaign	IntegrationId
	EstimatedCloseDate	ExternalSystemId
	Rating	LeadId
	Source	
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
	OpportunityId	
Opportunity	Opportunity	IntegrationId
	OpportunityId	ExternalSystemId
	Owner	OpportunityId
	Revenue	
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
	Account	

Table 170. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Personal Address	PersonalAddressIntegrationId	ExternalSystemId
	PersonalAddressName	
	AlternateCity	
	AlternateCountry	
	AlternateZipCode	
	AlternateStateProvince	
	AlternateAddress	
	AlternateAddress2	
	AlternateAddress3	
	Id	
	IntegrationId	
	AddressName	
	City	
	Country	
	ZipCode	
	StateProvince	
	Address	
	ShippingAddress2	
Related Contact	ContactRelationshipId	IntegrationId
	RelatedContactId	RelatedContactExternalId
	RelatedContactExternalId	
	RelatedContactIntegrationId	
	RelationshipStatus	
	RelationshipType	
	StartDate	
	EndDate	
	Description	
	RelationshipRole	
	ReverseRelationshipRole	

Table 170. Filterable Fields and User Key Fields on the Contact Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Revenue	ContactFullName	RevenueId
	Description	ExternalId
	ExternalId	IntegrationId
	Forecast	ExternalSystemId
	Frequency	
	IntegrationId	
	ModifiedDate	
	NumberOfPeriods	
	Product	
	ProductId	
	ProductCategoryId	
	ProductCategory	
	ProductCategoryExternalId	
	ProductCategoryIntegrationId	
	ProductExternalId	
	ProductIntegrationId	
	PurchasePrice	
	Quantity	
	Revenue	
	RevenueId	
	Status	
	StartCloseDate	
	Type	

Table 170. Filterable Fields and User Key Fields on the Contact Object's Child Components

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

Table 171 details the picklists available for the contact object.

Table 171. Picklists Available for the Contact Object

Child Component	Field Name
Contact (parent)	ContactType
	LeadSource
	MrMrs
	BestTimeToCall
	CallFrequency
	CurrentInvestmentMix
	Degree
	ExperienceLevel
	Gender
	InvestmentHorizon
	LifeEvent
	MaritalStatus
	MarketPotential
	Objective
	OwnOrRent
	PrimaryGoal
	RiskProfile
	Route
	Segment
	Tier
Account	Call Frequency
	Route
	Status
	Type
Contact Team	TeamRole
Interests	Category
	Subjects

Table 171. Picklists Available for the Contact Object

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

[Table 172](#) details a number of contact object fields that you must not use for customer integrations.

Table 172. Contact Object Fields That You Must Not Use

Field Name
PartyTypeCode
PartyUId
PersonUId

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

Current User

The current user object stores information on the currently logged-in user.

Child Components

[Login History](#) and [Quota](#)

Methods Called

[Table 173](#) details the method called by the Current user service, and its name as defined on the service.

Table 173. Methods Called by Current User Service

Method	Name as Defined in Service
"QueryPage" on page 91	CurrentUserQueryPage

Fields

[Table 174](#) details the required and read-only fields for the current user object.

Table 174. Required and Read-Only Fields for the Current User Object

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

[Table 175](#) 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 175. Filterable Fields on the Current User Object's Child Components

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

For more information on the fields accessible through the Current user Web service, go to the Web Services Administration page within the Oracle CRM On Demand application and generate the WSDL file for the current user object.

Related Topic

[User](#)

CustomObject1 - CustomObject3

The CustomObject1, CustomObject2, and CustomObject3 services expose the functionality of the CustomObject1 - CustomObject3 objects to external applications.

TIP: The reference information for each of CustomObject1, CustomObject2, and CustomObject3 follows the same pattern. In this topic, the information for CustomObject1 is given as an example.

NOTE: To download the CustomObject1 - CustomObject3 WSDL file, you must be given access to the relevant CustomObject object. If you do not have access to the CustomObject object, it is not available to download from the Web Services Administration page or available to use Web service calls. For assistance in gaining access to the CustomObject1 - CustomObject3 objects, contact your Oracle CRM On Demand service provider.

Parent Objects

[Account](#), [Contact](#)

Child Components

[Account](#), [Attachment](#), [Book](#), [Contact](#), [Opportunity](#), [Portfolio](#), and [Team](#).

CustomObject1, CustomObject2, and CustomObject3 also have child components for the other CustomObjects as follow:

- **CustomObject1.** CustomObject2 and CustomObject3 child components.
- **CustomObject2.** CustomObject1 and CustomObject3 child components.
- **CustomObject3.** CustomObject1 and CustomObject2 child components.

For information about using attachments with these objects, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

NOTE: For CustomObject3 only, the following fields are accessible: CustomObject5Id, CustomObject5Name, CustomObject5ExternalSystemId.

Methods Called

[Table 176](#) details the methods called by the CustomObject1 service. The methods for CustomObject2 and CustomObject3 follow the same pattern.

Table 176. Methods Called by CustomObject1 Service

Method	Name as Defined in Service
"Delete" on page 85	CustomObject1Delete
"DeleteChild" on page 88	CustomObject1DeleteChild
"Insert" on page 89	CustomObject1Insert
"InsertChild" on page 90	CustomObject1InsertChild

Table 176. Methods Called by CustomObject1 Service

Method	Name as Defined in Service
"InsertOrUpdate" on page 90	CustomObject1OrUpdate
"QueryPage" on page 91	CustomObject1Page
"Update" on page 101	CustomObject1Update
"UpdateChild" on page 102	CustomObject1UpdateChild

Fields

[Table 177](#) details the required and read-only fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 177. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
CustomObject1 (parent)	ParentId	Required
	ParentExternalSystemID	Required
	ParentIntegrationId	Required
	ParentObject1Id	Read-only
Account	CObj1AccountCreatedById	Read-only
	CObj1AccountCreatedDate	Read-only
	CObj1AccountModifiedById	Read-only
	CObj1AccountModifiedDate	Read-only
	Region	Read-only
	AccountType	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	CustomObjectId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 177. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Contact	CObj1ContactCreatedById	Read-only
	CObj1ContactCreatedDate	Read-only
	CObj1ContactModifiedById	Read-only
	CObj1ContactModifiedDate	Read-only
	ContactFirstName	Read-only
	ContactLastName	Read-only
	ContactType	Read-only
CustomObject2	CObj1CustomObject2CreatedById	Read-only
	CObj1CustomObject2CreatedDate	Read-only
	CObj1CustomObject2ModifiedById	Read-only
	CObj1CustomObject2ModifiedDate	Read-only
	CustomObject2Id	Read-only
CustomObject3	CObj1CustomObject3CreatedById	Read-only
	CObj1CustomObject3CreatedDate	Read-only
	CObj1CustomObject3ModifiedById	Read-only
	CObj1CustomObject3ModifiedDate	Read-only
	CustomObject3Id	Read-only
Opportunity	AccountName	Read-only
	CObj1OpportunityCreatedById	Read-only
	CObj1OpportunityCreatedDate	Read-only
	CObj1OpportunityModifiedById	Read-only
	CObj1OpportunityModifiedDate	Read-only
	OpportunityName	Read-only
	Revenue	Read-only
	SalesStage	Read-only

Table 177. Required and Read-Only Fields for the CustomObject1 Object

Child Component	Field Name	Type
Portfolio	AccountNumber	Read-only
	CObj1PortfolioCreatedById	Read-only
	CObj1PortfolioCreatedDate	Read-only
	CObj1PortfolioModifiedById	Read-only
	CObj1PortfolioModifiedDate	Read-only
	Revenue	Read-only
Team	CustomObject1TeamId	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only

Table 178 details the status key for the CustomObject1 object. The status keys for CustomObject2 and CustomObject3 follow a similar pattern.

Table 178. Status Key for the CustomObject1 Object

Child Component	Field Name
CustomObject1 (parent)	Audit Fields
	CustomObject1Id
	ExternalSystemId
	IntegrationId
Account	CustomObject1AccountId
	CObj1AccountCreatedById
	CObj1AccountCreatedDate
	CObj1AccountModifiedById
	CObj1AccountModifiedDate
Attachment	Audit Fields
	Id
	CustomObjectId
Book	Audit Fields
	BookId
	ModId

Table 178. Status Key for the CustomObject1 Object

Child Component	Field Name
Contact	CustomObject1ContactId
	CObj1ContactCreatedById
	CObj1ContactCreatedDate
	CObj1ContactModifiedById
	CObj1ContactModifiedDate
CustomObject2	CustomObject2Id
	CObj1CustomObject2CreatedById
	CObj1CustomObject2CreatedDate
	CObj1CustomObject2ModifiedById
	CObj1CustomObject2ModifiedDate
CustomObject3	CustomObject3Id
	CObj1CustomObject3CreatedById
	CObj1CustomObject3CreatedDate
	CObj1CustomObject3ModifiedById
	CObj1CustomObject3ModifiedDate
Opportunity	OpportunityId
	CObj1OpportunityCreatedById
	CObj1OpportunityCreatedDate
	CObj1OpportunityModifiedById
	CObj1OpportunityCreatedDate
Portfolio	PortfolioId
	CObj1PortfolioCreatedById
	CObj1PortfolioCreatedDate
	CObj1PortfolioModifiedById
	CObj1PortfolioModifiedDate
Team	Audit Fields
	CustomObject1TeamId

Table 179 details the pick map fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 179. Pick Map Fields for the CustomObject1 Object

Child Component	Pick Map Field	Maps To
CustomObject1 (parent)	AccountExternalId	AccountId
	AccountIntegrationId	AccountId
	AccountName	AccountId
	ActivityExternalId	ActivityId
	ActivityIntegrationId	ActivityId
	ActivityName	ActivityId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	CampaignName	CampaignId
	ContactExternalId	ContactId
	ContactFirstName	ContactId
	ContactFullName	ContactId
	ContactIntegrationId	ContactId
	ContactLastName	ContactId
	CustomObject2ExternalId	CustomObject2Id
	CustomObject2IntegrationId	CustomObject2Id
	CustomObject2Name	CustomObject2Id
	CustomObject3ExternalId	CustomObject3Id
	CustomObject3IntegrationId	CustomObject3Id
	CustomObject3Name	CustomObject3Id
	DealerName	DealerId
	HouseholdExternalId	HouseholdId
	HouseholdIntegrationId	HouseholdId
	HouseholdName	HouseholdId
	LeadExternalId	LeadId
	LeadFirstName	LeadId
	LeadFullName	LeadId
	LeadIntegrationId	LeadId

Table 179. 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 179. Pick Map Fields for the CustomObject1 Object

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

Table 180 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 180. Filterable Fields and User Key Fields on the CustomObject1 Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
CustomObject1 (parent)	All	CustomObject1Id
		ExternalSystemId
		IntegrationId
Account	AccountId	CustomObject1AccountId
	AccountType	ExternalSystemId
	CObj1AccountModifiedById	IntegrationId
	CObj1AccountModifiedDate	
	ExternalSystemId	
	IntegrationId	
	Location	
	Name	
	Region	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 180. 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
	CObj1PortfolioModifiedDate	ExternalSystemId
	ExternalSystemId	IntegrationId
	IntegrationId	
	PortfolioId	

Table 180. 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 181](#) details the picklists available for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 181. Picklists Available for the CustomObject1 Object

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

Lead

The lead object stores information on a company or individual with whom an opportunity can be created. It allows the user to identify the companies that might be interested in a product or service. Leads are usually generated as part of a marketing campaign.

Parent Objects

[Account](#), [Campaign](#), [Contact](#), and [Opportunity](#)

Child Components

[Activity](#), [Attachment](#), [Book](#), [Campaign](#).

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Methods Called

Table 182 details the methods called by the Lead service.

Table 182. Methods Called by Lead Service

Method	Name as Defined in Service
"Delete" on page 85	LeadDelete
"DeleteChild" on page 88	LeadDeleteChild
"Insert" on page 89	LeadInsert
"InsertChild" on page 90	LeadInsertChild
"InsertOrUpdate" on page 90	LeadInsertOrUpdate
"QueryPage" on page 91	LeadQueryPage
"Update" on page 101	LeadUpdate
"UpdateChild" on page 102	LeadUpdateChild

Fields

Table 183 details the required and read-only fields for the lead object.

Table 183. Required and Read-Only Fields for the Lead Object

Child Component	Field Name	Type
Lead (parent)	FirstName	Required
	LastName	Required
	LeadOwner	Required
	ContactFullName	Read-only
	Audit Fields	Read-only
	LastUpdated	Read-only
	LeadConcatField	Read-only
	LeadFullName	Read-only
	ReferredById	Read-only

Table 183. Required and Read-Only Fields for the Lead Object

Child Component	Field Name	Type
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	Lead	Read-only
	LeadExternalId	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Duration	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	LeadId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 184 details the status key for the lead object.

Table 184. Status Key for the Lead Object

Child Component	Field Name
Lead (parent)	Audit Fields
	AccountId
	CampaignId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
	OpportunityId
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LeadId
Attachment	Audit Fields
	Id
	LeadId
Book	Audit Fields
	BookId
	ModId

Table 185 details the pick map fields for the lead object.

Table 185. Pick Map Fields for the Lead Object

Child Component	Pick Map Field	Maps To
Lead (parent)	Campaign	CampaignId
	OpportunityName	OpportunityId
	Owner	OwnerId
	AccountExternalSystemId	AccountId
	OpportunityExternalSystemId	OpportunityId
	ContactExternalSystemId	ContactId
	CampaignExternalSystemId	CampaignId
	ReferredByExternalSystemId	ReferredById
Activity	AccountIntegrationId	AccountId
	AccountExternalId	AccountId
	Address	AddressId
	AssignedTo	AssignedToId
	CampaignExternalId	CampaignId
	CampaignIntegrationId	CampaignId
	DelegatedBy	DelegatedById

Table 185. 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 186 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 186. Filterable Fields and User Key Fields on the Lead Object's Child Components

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

Table 186. Filterable Fields and User Key Fields on the Lead Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity	Type	Type and Description
	Owner	IntegrationId
	Subject	ExternalSystemId
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Partner	LeadPartnerId	PartnerIntegrationId
	PartnerId	PartnerExternalSystemId
	PartnerExternalSystemId	
	CreatedDate	
	ModifiedDate	

Table 187 details the picklists available for the lead object.

Table 187. Picklists Available for the Lead Object

Field Name
Country
MrMrs
Rating
Source
StateProvince
Status

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

Note

The note object stores information about the notes available in the Message Center in the Oracle CRM On Demand application. The notes can be sent from users or can store extra information (as a note) on a parent object. This allows employees who are working on a particular record to add extra information as they see fit. For example, when talking to a contact, an employee might notice that the contact is not happy with a service provided. The employee can record this information in a note so that any other employees who talk to the contact are aware of the contact's dissatisfaction.

The note object has no child components.

Parent Objects

[Account](#), [Campaign](#), [Contact](#), [Opportunity](#), and [Service Request](#)

Methods Called

[Table 188](#) details the methods called by the Note service.

Table 188. Methods Called by Note Service

Method	Name as Defined in Service
"Delete" on page 85	NoteDelete
"Insert" on page 89	NoteInsert
"InsertOrUpdate" on page 90	NoteInsertOrUpdate
"QueryPage" on page 91	NoteQueryPage
"Update" on page 101	NoteUpdate

Fields

Table 189 details the required and read-only fields for the note object.

Table 189. Required and Read-Only Fields for the Note Object

Child Component	Field Name	Type
Note (parent)	Subject	Required
	NotelId	Read-only
	OwnerId	Read-only
	OwnerAlias	Read-only
	ParentNotelId	Read-only
	SourceId	Read-only
	SourceName	Read-only
	Audit Fields	Read-only

Table 190 details the status key for the note object.

Table 190. Status Key for the Note Object

Child Component	Field Name
Note (parent)	Audit Fields
	NotelId

Table 191 provides a list of the filterable fields and a list of user key combinations for the note object.

Table 191. Filterable Fields and User Key Fields on the Note Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Note (parent)	All	NotelId

For more information on the fields accessible through the Note Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the note object.

Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Parent Objects

[Account](#), [Campaign](#), [Contact](#), [CustomObject1](#) - [CustomObject3](#)

Child Components

[Activity](#), [Attachment](#), [Book](#), [Competitor](#), [Contact](#), [Lead](#), [Note](#), and [OpportunityTeam](#).

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

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Methods Called

[Table 192](#) details the methods called by the Opportunity service.

Table 192. Methods Called by Opportunity Service

Method	Name as Defined in Service
"Delete" on page 85	OpportunityDelete
"DeleteChild" on page 88	OpportunityDeleteChild
"Insert" on page 89	OpportunityInsert
"InsertChild" on page 90	OpportunityInsertChild
"InsertOrUpdate" on page 90	OpportunityInsertOrUpdate
"QueryPage" on page 91	OpportunityQueryPage
"Update" on page 101	OpportunityUpdate
"UpdateChild" on page 102	OpportunityUpdateChild

Fields

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

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

Child Component	Field Name	Type
Opportunity (parent)	AccountId	Required
	CloseDate	Required
	OpportunityName	Required
	SalesStage	Required
	Audit Fields	Read-only
	LastUpdated	Read-only
	OpportunityConcatField	Read-only
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	Opportunity	Read-only
	OpportunityIntegrationId	Read-only
	OpportunityExternalId	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTime	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Duration	Read-only
	Audit Fields	Read-only

Table 193. 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 193. 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 194 details the status key for the opportunity object.

Table 194. Status Key for the Opportunity Object

Child Component	Field Name
Opportunity (parent)	Audit Fields
	AccountId
	ExternalSystemId
	Id
	IntegrationId
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	OpportunityId
Attachment	Audit Fields
	Id
	OpportunityId
Book	Audit Fields
	BookId
	ModId
Competitor	Audit Fields
	OpportunityCompetitorId
	CompetitorId
	CompetitorExternalSystemId
Contact	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	OpportunityId

Table 194. 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 195](#) details the pick map fields for the opportunity object.

Table 195. Pick Map Fields for the Opportunity Object

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

Table 195. 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	AccountExternalSystemId	AccountId
	AccountIntegrationId	AccountId
	ManagerExternalSystemId	ManagerId
	ManagerIntegrationId	ManagerId
	OwnerEmailAddress	OwnerId
	OwnerExternalId	OwnerId
	OwnerIntegrationId	OwnerId
	SourceCampaignName	SourceCampaignId

Table 195. Pick Map Fields for the Opportunity Object

Child Component	Pick Map Field	Maps To
	SourceCampaignExternalId	SourceCampaignId
	TimeZoneName	TimeZoneId
	CustomObject1Name	CustomObject1Id
	CustomObject2Name	CustomObject2Id
	CustomObject3Name	CustomObject3Id
Competitor	PrimaryContactName	ContactId
	PartnerExternalSystemId	PartnerId
	PartnerName	PartnerId
Partner	PrimaryContactName	ContactId
	CompetitorExternalSystemId	CompetitorId
	CompetitorName	CompetitorId
Product	ProductName	ProductId
	ProductExternalSystemId	ProductId
	ProductIntegrationId	ProductId
	ContactExternalSystemId	ContactId
	ContactIntegrationId	ContactId
	Owner	OwnerId
	CustomObject5Name	CustomObject5Id
	CustomObject5ExternalSystemId	CustomObject5Id
	CustomObject6Name	CustomObject6Id
	CustomObject6ExternalSystemId	CustomObject6Id

Table 196 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 196. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Opportunity (parent)	All	OpportunityId
		IntegrationId
		ExternalSystemId

Table 196. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Activity	Type	Type and Description
	Owner	IntegrationId
	Subject	ExternalSystemId
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName
Competitor	OpportunityCompetitorId	OpportunityCompetitorId
	PrimaryContactId	CompetitorExternalSystemId
	EndDate	CompetitorId
	CompetitorId	
	CompetitorExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	
Contact	ContactType	AccountName and Private
	ContactFirstName	ContactFirstName and ContactLastName and Private
	JobTitle	IntegrationId
	ContactLastName	ExternalSystemId
	Owner	
	Id	

Table 196. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Lead	Campaign	IntegrationId
	EstimatedCloseDate	ExternalSystemId
	Rating	
	Source	
	Status	
	LeadOwner	
	PotentialRevenue	
	ProductInterest	
	SalesPerson	
	LeadId	
Note	Subject	Subject and Description
		IntegrationId
		ExternalSystemId
OpportunityTeam	UserFirstName	OpportunityTeamId
	UserLastName	UserId
	ModifiedDate	UserExternalSystemId
	OpportunityAccess	UserEmail
	OpportunityAccessId	UserAlias
	OpportunityTeamId	
	TeamRole	
	UserExternalSystemId	
	UserId	
Partner	OpportunityPartnerId	OpportunityPartnerId
	PrimaryContactId	PartnerExternalSystemId
	EndDate	PartnerId
	PartnerId	
	PartnerExternalSystemId	
	ReverseRelationshipRole	
	RelationshipRole	
	StartDate	
	ModifiedDate	

Table 196. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Product	OpportunityIntegrationId	ExternalSystemId
	OpportunityExternalSystemId	OpportunityIntegrationId
	OpportunitySalesStage	ProductRevenueId
	OpportunityAccountId	IntegrationId
	OpportunityAccountName	
	OpportunityAccountLocation	
	OpportunityAccountExternalSystemId	
	OpportunityAccountIntegrationId	
	ModifiedDate	
	ContactId	
	ContactExternalSystemId	
	ContactIntegrationId	
	Contract	
	OwnerId	
	Owner	
	SerialNumber	
	Revenue	
	ExpectedRevenue	
	Quantity	
	PurchasePrice	
	PurchaseDate	

Table 196. Filterable Fields and User Key Fields on the Opportunity Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
	StartCloseDate	
	NumberOfPeriods	
	Frequency	
	Probability	
	Forecast	
	AssetValue	
	Premium	
	ShipDate	
	Status	
	Type	
	Warranty	
	ProductRevenueId	
	ExternalSystemId	
	IntegrationId	
	ProductId	
	ProductName	
	ProductExternalSystemID	
	ProductionIntegrationId	
	ProductCategoryId	
	ProductCategory	
	ProductPartNumber	
	ProductStatus	
	ProductType	
	OpportunityId	
	OpportunityName	

Table 197 details the picklists available for the opportunity object.

Table 197. Picklists Available for the Opportunity Object

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

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

Product

The product object allows you to define and record details about a product or service that your company sells to its customers, including information on product price, category, and so on. The product object does not have any child objects.

Parent Objects

[Account](#), [Campaign](#) and [Contact](#)

Methods Called

Table 198 details the methods called by the Product service.

Table 198. Methods Called by Product Service

Method	Name as Defined in Service
"Insert" on page 89	ProductInsert
"InsertOrUpdate" on page 90	ProductInsertOrUpdate
"QueryPage" on page 91	ProductQueryPage
"Update" on page 101	ProductUpdate

Fields

All fields on the product object are filterable.

Table 199 details the required and read-only fields for the product object.

Table 199. Required and Read-Only Fields for the Product Object

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

Table 200 details the status key for the product object.

Table 200. Status Key for the Product Object

Child Component	Field Name
Product (parent)	Audit Fields
	Id
	IntegrationId

Table 201 details the pick map field for the product object.

Table 201. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 202 details the user keys for the product object.

Table 202. User Keys for the Product Object

Child Component	Field Name
Product (parent)	ProductId
	IntegrationId
	ExternalSystemId

Table 203 details the picklists available for the product object.

Table 203. Picklists Available for the Product Object

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

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

Related Topic

[Product Category](#)

Product Category

The product category object allows you to logically sort products into groups, where each product is in some way related to the other products in the category. The product category object does not have any child objects.

Parent Objects

None

Methods Called

[Table 204](#) details the methods called by the Product category service.

Table 204. Methods Called by Product Category Service

Method	Name as Defined in Service
"Delete" on page 85	ProductCategoryDelete
"Insert" on page 89	ProductCategoryInsert
"InsertOrUpdate" on page 90	ProductCategoryInsertOrUpdate
"QueryPage" on page 91	ProductCategoryQueryPage
"Update" on page 101	ProductCategoryUpdate

Fields

All fields on the product category object are filterable.

[Table 205](#) details the required and read-only fields for the product category object.

Table 205. 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 206](#) details the status key for the product category object.

Table 206. Status Key for the Product Category Object

Child Component	Field Name
ProductCategory	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	Name

[Table 207](#) details the pick map field for the product category object.

Table 207. Pick Map Field for the Product Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

[Table 208](#) details the user keys for the product category object.

Table 208. User Keys for the Product Category Object

Child Component	Field Name
ProductCategory	ProductCategoryId
	IntegrationId
	ExternalSystemId
	Name

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

Related Topic

[Automotive Edition Parent Objects \(Web Services v1.0\)](#)

Service Request

The service request object allows customers to request information or assistance with a problem related to products or services purchased from your company. Service requests can be ranked for severity and prioritized accordingly.

Parent Objects

[Account](#), [Contact](#), and [Solution](#)

Child Components

[Activity](#), [Attachment](#), [Audit Trail](#), [Book](#), [Note](#), and [Solution](#).

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Methods Called

[Table 209](#) details the methods called by the Service request service.

Table 209. Methods Called by Service Request Service

Method	Name as Defined in Service
"Delete" on page 85	ServiceRequestDelete
"DeleteChild" on page 88	ServiceRequestDeleteChild
"Insert" on page 89	ServiceRequestInsert
"InsertChild" on page 90	ServiceRequestInsertChild
"InsertOrUpdate" on page 90	ServiceRequestInsertOrUpdate
"QueryPage" on page 91	ServiceRequestQueryPage
"Update" on page 101	ServiceRequestUpdate
"UpdateChild" on page 102	ServiceRequestUpdateChild

Fields

Table 210 details the required and read-only fields for the service request object.

Table 210. Required and Read-Only Fields for the Service Request Object

Child Component	Field Name	Type
ServiceRequest (parent)	ContactEmail	Read-only
	ContactFirstName	Read-only
	ContactFullName	Read-only
	ContactLastName	Read-only
	Audit Fields	Read-only
	LastUpdated	Read-only
	ServiceRequestConcatId	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	SRId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 211 details the status key for the service request object.

Table 211. Status Key for the Service Request Object

Child Component	Field Name
ServiceRequest (parent)	Audit Fields
	AccountId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Activity	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
Attachment	Audit Fields
	Id
	SRId
Book	Audit Fields
	BookId
	ModId
ServiceRequestNote	Audit Fields
	ExternalSystemId
	Id
	LastUpdated
	ServiceRequestId
Solution	Audit Fields
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 212 details the pick map fields for the service request object.

Table 212. Pick Map Fields for the Service Request Object

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

Table 213 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 213. Filterable Fields and User Key Fields on the Service Request Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Service Request (parent)	All	ServiceRequestId
		IntegrationId
		ExternalSystemId
		SRNumber
Activity	Type	Type and Description
	Owner	IntegrationId
	Subject	ExternalSystemId
	DueDate	
	Priority	
	Status	
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Audit Trail	Date	None
	User	
	FieldModified	

Table 213. Filterable Fields and User Key Fields on the Service Request Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Book	None	BookId
		BookName
Service Request Note	Subject	Subject and Description
		ExternalSystemId
Solution	Title	Title
	Published	IntegrationId
	SolutionId	ExternalSystemId
	Status	
	Id	

[Table 214](#) details the picklists available for the service request object.

Table 214. Picklists Available for the Service Request Object

Field Name
Area
Cause
Priority
Source
Status
Type

For more information on the fields accessible through the Service request Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the service request object.

Solution

The solution object stores information on solutions to customer problems or service requests. Solutions can be reused if the same problem is identified with a product or service. This prevents the duplication of work for customer service representatives.

Parent Object

[Activity](#) and [Service Request](#)

Child Component

[Attachment](#) and [Service Request](#).

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

Methods Called

[Table 215](#) details the methods called by the Solution service.

Table 215. Methods Called by Solution Service

Method	Name as Defined in Service
"Delete" on page 85	SolutionDelete
"DeleteChild" on page 88	SolutionDeleteChild
"Insert" on page 89	SolutionInsert
"InsertChild" on page 90	SolutionInsertChild
"InsertOrUpdate" on page 90	SolutionInsertOrUpdate
"QueryPage" on page 91	SolutionQueryPage
"Update" on page 101	SolutionUpdate
"UpdateChild" on page 102	SolutionUpdateChild

Fields

[Table 216](#) details the required and read-only fields for the solution object.

Table 216. Required and Read-Only Fields for the Solution Object

Child Component	Field Name	Type
Solution (parent)	Title	Required
	Audit Fields	Read-only
	CreatorId	Read-only
	LastUpdated	Read-only

Table 216. Required and Read-Only Fields for the Solution Object

Child Component	Field Name	Type
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	SolutionId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only

[Table 217](#) details the status key for the solution object.

Table 217. Status Key for the Solution Object

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

Table 218 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 218. Filterable Fields and User Key Fields on the Solution Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Solution (parent)	All	SolutionId
		IntegrationId
		ExternalSystemId
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Service Request	Subject	SRNumber
	Area	IntegrationId
	Owner	ExternalSystemId
	Priority	
	Type	
	Cause	
	Source	
	Status	

Table 219 details the picklists available for the solution object.

Table 219. Picklists Available for the Solution Object

Field Name
Area
Cause
Priority
Source
Status
Type

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

Related Topic

[Service Request](#)

Territory

The territory object allows you to store information about the sales territory that is assigned to a user. This information includes the territory name, a description, the currency code, and the sales quota for the territory. The territory object does not have any associated child objects or parent objects.

Parent Objects

None

Methods Called

[Table 220](#) details the methods called by the Territory service.

Table 220. Methods Called by Territory Service

Method	Name as Defined in Service
"Delete" on page 85	TerritoryDelete
"Insert" on page 89	TerritoryInsert
"InsertOrUpdate" on page 90	TerritoryInsertOrUpdate
"QueryPage" on page 91	TerritoryQueryPage
"Update" on page 101	TerritoryUpdate

Fields

All fields on the territory object are filterable. The TerritoryName, TerritoryExternalSystemId, and TerritoryIntegrationId fields are user keys for the territory object.

[Table 221](#) details the required and read-only fields for the territory object.

Table 221. Required and Read-Only Fields for the Territory Object

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

Table 222 details the status key for the territory object.

Table 222. Status Key for the Territory Object

Child Component	Field Name
Territory (parent)	Audit Fields
	TerritoryExternalSystemId
	TerritoryIntegrationId
	TerritoryId

Table 223 details the pick map field for the territory object.

Table 223. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

Table 224 details the picklists available for the territory object.

Table 224. Picklists Available for the Territory Object

Field Name
ParentTerritoryExternalSystemId
ParentTerritoryIntegrationId

For more information on the fields accessible through the Territory Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the territory object.

User

The user object allows you to define and record details of all users in Oracle CRM On Demand, for example, name, position, contact details, manager, and so on. It is different from the [Current User](#) object in that it is not restricted only to the currently logged in user. It enables queries to be run on all users, and enables an administrator to insert and update a user's profile. The user object does not have any child components.

Usage

The UserLoginId and UserSignInId fields must be used as follows:

- **UserLoginId.** Used for creating user records through the User Web service.
- **UserSignInId.** Used as the user name for logging in and authenticating using Web services. Also, used for queries, as using UserLoginId is not allowed for queries.

Parent Object

[User Group](#)

Child Object

[Address](#), [DelegatedUser](#)

Methods Called

[Table 225](#) details the methods called by the User service.

Table 225. Methods Called by User Service

Method	Name as Defined in Service
"DeleteChild" on page 88	UserDeleteChild
"Insert" on page 89	UserInsert
"InsertChild" on page 90	UserInsertChild
"InsertOrUpdate" on page 90	UserInsertOrUpdate
"QueryPage" on page 91	UserQueryPage
"Update" on page 101	UserUpdate
"UpdateChild" on page 102	UserUpdateChild

Fields

All fields on the user object are filterable.

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

Table 226. 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 227 details the status key for the user object.

Table 227. Status Key for the User Object

Child Component	Field Name
User	ModifiedById
	ModifiedDate
	EMailAddr
	UserId
	IntegrationId

Table 228 details the pick map field for the user object.

Table 228. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

[Table 229](#) provides a list of user key combinations for the user object.

Table 229. User Key Fields on the User Object

Child Components	User Key Field Combinations
User	UserId
	ExternalSystemId
	IntegrationId
	EmailAddr
Address	IntegrationId
DelegatedUser	ExternalSystemId

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

Related Topic

[Current User](#)

User Group

The User Group object allows you to create groups to which users can be added. Users can only be a member of one group, and groups can contain many users.

Parent Objects

None

Child Component

[User](#)

Methods Called

Table 230 details the methods called by the User group service.

Table 230. Methods Called by User Group Service

Method	Name as Defined in Service
"Delete" on page 85	UserGroupDelete
"DeleteChild" on page 88	UserGroupDeleteChild
"Insert" on page 89	UserGroupInsert
"InsertChild" on page 90	UserGroupInsertChild
"InsertOrUpdate" on page 90	UserGroupInsertOrUpdate
"QueryPage" on page 91	UserGroupQueryPage
"Update" on page 101	UserGroupUpdate
"UpdateChild" on page 102	UserGroupUpdateChild

Fields

Table 231 details the required and read-only fields for the user group object.

Table 231. Required and Read-Only Fields for the User Group Object

Child Component	Field Name	Type
User Group (parent)	Name	Required
	UserGroupId	Read-only
	Audit Fields	Read-only
User	UserGroupUserId	Read-only
	UserId	Read-only
	Alias	Read-only
	Email	Read-only
	Role	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only
	Audit Fields	Read-only

Table 232 details the status key for the user group object.

Table 232. Status Key for the User Group Object

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

Table 233 details the pick map field for the user group object.

Table 233. Pick Map Field for the User Group Object

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

Table 234 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 234. Filterable Fields and User Key Fields on the User Group Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
User Group (parent)	All	Name
		UserGroupIntegrationId
		UserGroupExternalSystemId

Table 234. Filterable Fields and User Key Fields on the User Group Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
User	UserGroupId	UserIntegrationId
	UserId	UserExternalSystemId
	UserIntegrationId	
	UserExternalSystemId	
	Alias	
	Email	
	Role	
	UserFirstName	
	UserLastName	

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

Life Sciences Edition Parent Objects (Web Services v1.0)

This topic includes the parent objects available with Oracle CRM On Demand Life Sciences Edition.

To download WSDL files for these objects, you must be given access to the object. If you do not have access to the object, it is not available to download from the Web Services Administration page or available to use the Web service calls. For assistance in gaining access to the object, contact your Oracle CRM On Demand service provider.

The following objects are detailed in this topic:

■ [“MedEd” on page 405](#)

MedEd

The MedEd object allows you to plan and track medical education events. A medical education event can be as simple as a lunch-and-learn session in a physician's office or as complex as a seminar series or national sales meeting.

Child Component

[Invitee](#)

Methods Called

Table 235 details the methods called by the MedEd service.

Table 235. Methods Called by MedEd Service

Method	Name as Defined in Service
"Delete" on page 85	MedEdDelete
"DeleteChild" on page 88	MedEdDeleteChild
"Insert" on page 89	MedEdInsert
"InsertChild" on page 90	MedEdInsertChild
"InsertOrUpdate" on page 90	MedEdInsertOrUpdate
"QueryPage" on page 91	MedEdQueryPage
"Update" on page 101	MedEdUpdate
"UpdateChild" on page 102	MedEdUpdateChild

Fields

Table 236 details the read-only fields for the MedEd object and its child component.

Table 236. Read-Only Fields on the MedEd Object

Child Component	Field Name	Type
MedEd (parent)	EndDate	Required
	Name	Required
	Objective	Required
	StartDate	Required
	Audit Fields	Read-only
Invitee	Status	Required
	InviteeId	Read-only
	Audit Fields	Read-only

Table 237 details the status key for the MedEd object.

Table 237. Status Key for the MedEd Object

Child Component	Field Name
MedEd (parent)	Audit Fields
	ExternalId
	MedEdId
Invitee	Audit Fields
	InviteeExternalSystemId
	InviteeId

Table 238 details the pickmap fields for the MedEd object and its child objects.

Table 238. Pick Map Fields for the MedEd Object

Child Component	Pick Map Field	Maps To
MedEd (parent)	ProductExternalId	ProductId
	ProductIntegrationId	ProductId

Table 239 provides a list of the filterable fields and user key combinations for the child components of the MedEd object.

Table 239. Filterable Fields and User Key Fields on the MedEd Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
MedEd (parent)	ExternalSystemId	MedEdId
	ProductIntegrationId	ExternalSystemId
	ProductId	
	ProductId	
	ProductExternalId	
	PrimaryOwnerId	
Invitee	ContactId	InviteeId
	InviteeExternalSystemId	InviteeExternalSystemId
	Status	
	Type	
	ModifiedDate	

[Table 240](#) details the picklists available for the MedEd object.

Table 240. Picklists Available for the MedEd Object

Child Component	Field Name
MedEd (parent)	EventStatusCode
	EventTypeCode
Invitee	InviteeStatus

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

Related Topic

[Invitee](#)

Financial Services Edition Parent Objects (Web Services v1.0)

This topic includes the parent objects available with Oracle CRM On Demand Financial Services Edition.

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

The following objects are detailed in this topic:

- ["Household" on page 408](#)
- ["Portfolio" on page 413](#)

Household

The household object allows you to define and record financial details about a group of contacts that live in the same household, for example, parents, brothers, sisters, spouses, and so on. These details include the assets of the household, the liabilities of the household, the net income of the household, and so on.

Child Component

[Book](#), [HouseholdTeam](#)

Methods Called

Table 241 details the methods called by the Household service.

Table 241. Methods Called by Household Service

Method	Name as Defined in Service
"Delete" on page 85	HouseholdDelete
"DeleteChild" on page 88	HouseholdDeleteChild
"Insert" on page 89	HouseholdInsert
"InsertChild" on page 90	HouseholdInsertChild
"InsertOrUpdate" on page 90	HouseholdInsertOrUpdate
"QueryPage" on page 91	HouseholdQueryPage
"Update" on page 101	HouseholdUpdate
"UpdateChild" on page 102	HouseholdInsertChild

Fields

[Table 242](#) details the required and read-only fields for the household object.

Table 242. Required and Read-Only Fields for the Household Object

Child Component	Field Name	Type
Household (parent)	HouseholdName	Required
	IntegrationID	Required
	ExternalSystemID	Required
	HouseholdId	Read-only
	PrimaryContactId	Read-only
	PrimaryContactExternalId	Read-only
	PrimaryContactIntegrationId	Read-only
	PrimaryContactFirstName	Read-only
	PrimaryContactLastName	Read-only
	Timezone	Read-only
	HouseholdCurrency	Read-only
	LastActivity	Read-only
	HeadDOB	Read-only
	TotalIncome	Read-only
	TotalAssets	Read-only
	TotalExpenses	Read-only
	TotalLiabilities	Read-only
	TotalNetWorth	Read-only
	RiskProfile	Read-only
	ExperienceLevel	Read-only
	InvestmentHorizon	Read-only
	CurrentInvestmentMix	Read-only
	Objective	Read-only
	PrimaryGoal	Read-only
	Audit Fields	Read-only

Table 242. 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 243](#) details the status key for the household object.

Table 243. Status Key for the Household Object

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

Table 243. Status Key for the Household Object

Child Component	Field Name
HouseholdTeam	Audit Fields
	UserExternalSystemId
	HouseholdTeamId
	UserAlias
	UserEmail

[Table 244](#) details the pick map fields for the household object.

Table 244. 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 245](#) 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 245. Filterable Fields and User Key Fields on the Household Object's Child Components

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

Table 245. Filterable Fields and User Key Fields on the Household Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact	ContactID	ContactID
	ContactExternalId	ContactExternalId
	ContactIntegrationId	ContactIntegrationId
	ModifiedDate	
	RelationshipRole	
HouseholdTeam	UserId	HouseholdTeamId
	UserExternalSystemId	UserExternalSystemId
	LastName	UserAlias
	FirstName	UserEmail
	TeamRole	
	HouseholdAccess	
	HouseholdTeamId	

Table 246 details the picklists available for the household object.

Table 246. Picklists Available for the Household Object

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

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

Portfolio

The portfolio object allows you to define and record details about the collection of financial services that you can provide to an account. Financial services include loans, credit cards, insurance, general banking, and so on.

Child Component

[Book](#), [Contact](#), [PortfolioTeam](#)

Methods Called

[Table 247](#) details the methods called by the Portfolio service.

Table 247. Methods Called by Portfolio Service

Method	Name as Defined in Service
"Delete" on page 85	PortfolioDelete
"DeleteChild" on page 88	PortfolioDeleteChild
"Insert" on page 89	PortfolioInsert
"InsertChild" on page 90	PortfolioInsertChild
"InsertOrUpdate" on page 90	PortfolioInsertOrUpdate
"QueryPage" on page 91	PortfolioQueryPage
"Update" on page 101	PortfolioUpdate
"UpdateChild" on page 102	PortfolioUpdateChild

Fields

[Table 248](#) details the read-only fields for the portfolio object and its child component.

Table 248. Read-Only Fields on the Portfolio Object

Child Component	Field Name	Type
Portfolio (parent)	PortfolioId	Read-only
	Owner	Read-only
	OwnerId	Read-only
	PrimaryContact	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 248. 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 249](#) details the status key for the portfolio object.

Table 249. Status Key for the Portfolio Object

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

Table 250 details the pickmap fields for the portfolio object and its child objects.

Table 250. Pick Map Fields for the Portfolio Object

Child Component	Pick Map Field	Maps To
Portfolio (parent)	InstitutionExternalId	InstitutionId
	InstitutionIntegrationId	InstitutionId
	InstitutionName	InstitutionId
	InstitutionLocation	InstitutionId
	Product	ProductId
	ProductExternalId	ProductId
	ProductIntegrationId	ProductId
Book	BookName	BookId
Contact	ContactExternalId	ContactId
	ContactIntegrationId	ContactId
PortfolioTeam	UserEmail	UserId
	UserAlias	UserId
	UserExternalSystemId	UserId
	LastName	UserId
	FirstName	UserId
	FullName	UserId

Table 251 provides a list of the filterable fields and user key combinations for the child components of the portfolio object.

Table 251. Filterable Fields and User Key Fields on the Portfolio Object's Child Components

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

Table 251. 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 252 details the picklists available for the portfolio object.

Table 252. Picklists Available for the Portfolio Object

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

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

Automotive Edition Parent Objects (Web Services v1.0)

This topic includes the parent objects available with Oracle CRM On Demand Automotive Edition.

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

The following objects are detailed in this topic:

- [“Dealer” on page 418](#)
- [“Vehicle” on page 421](#)

Dealer

The dealer object stores information about dealerships in the automotive industry, for example, the name of the dealership, the identity of the parent dealership, the site on which the dealership is based, and so on. The dealer object does not have any parent objects.

Child Objects

[Attachment](#) and [Book](#).

For information about using attachments with this object, see [Appendix A, “Using Attachments with Web Services On Demand.”](#)

Methods Called

[Table 253](#) details the methods called by the Dealer service.

Table 253. Methods Called by Dealer Service

Method	Name as Defined in Service
“Delete” on page 85	DealerDelete
“DeleteChild” on page 88	DealerDeleteChild
“Insert” on page 89	DealerInsert
“InsertChild” on page 90	DealerInsertChild
“InsertOrUpdate” on page 90	DealerInsertOrUpdate
“QueryPage” on page 91	DealerQueryPage
“Update” on page 101	DealerUpdate
“UpdateChild” on page 102	DealerUpdateChild

Fields

All fields on the dealer object are filterable.

[Table 254](#) details the required and read-only fields for the dealer object.

Table 254. Required and Read-Only Fields for the Dealer Object

Child Component	Field Name	Type
Dealer	DealerId	Required
	DealerIntegrationId	Required
	DealerExternalSystemID	Required
	DealerId	Read-only
	DealerType	Read-only
	Audit Fields	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	DealerId	Read-only
	Id	Read-only
	ModId	Read-only
	Audit Fields	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	Audit Fields	Read-only

Table 255 details the status key for the dealer object.

Table 255. Status Key for the Dealer Object

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

Table 256 details the pick map fields for the dealer object.

Table 256. Pick Map Fields for the Dealer Object

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

Table 257 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 257. Filterable Fields and User Key Fields on the Dealer Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Dealer (parent)	All	DealerId
		DealerIntegrationID
		DealerExternalSystemID

Table 257. Filterable Fields and User Key Fields on the Dealer Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Attachment	None	Id
		ExternalSystemId
		FileNameOrURL and FileExtension
Book	None	BookId
		BookName

Table 258 details the picklists available for the dealer object.

Table 258. Picklists Available for the Dealer Object

Field Name
ParentDealerName
ParentDealerSite

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

Vehicle

The vehicle object allows you to create and store information about a vehicle, for example, a car, a truck, a van, and so on, that your company would like to sell to a contact or account. This information includes the vehicle's current mileage, the invoice price, the dealership, the make, and so on.

Parent Objects

None

Child Component

[Contact](#)

Methods Called

Table 259 details the methods called by the Vehicle service.

Table 259. Methods Called by Vehicle Service

Method	Name as Defined in Service
"Delete" on page 85	VehicleDelete
"DeleteChild" on page 88	VehicleDeleteChild
"Insert" on page 89	VehicleInsert
"InsertChild" on page 90	VehicleInsertChild
"InsertOrUpdate" on page 90	VehicleInsertOrUpdate
"QueryPage" on page 91	VehicleQueryPage
"Update" on page 101	VehicleUpdate
"UpdateChild" on page 102	VehicleUpdateChild

Fields

Table 260 details the required and read-only fields for the vehicle object.

Table 260. Required and Read-Only Fields for the Vehicle Object

Child Component	Field Name	Type
Vehicle (parent)	VehicleId	Read-only
	Contact	Read-only
	ProductType	Read-only
	SellingDealer	Read-only
	ServicingDealer	Read-only
	Audit Fields	Read-only
Contact	ContactId	Required
	ContactExternalSystemId	Required
	ContactIntegrationId	Required
	Audit Fields	Read-only

Table 261 details the status key for the vehicle object.

Table 261. Status Key for the Vehicle Object

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

Table 262 details the pick map fields for the vehicle object.

Table 262. Pick Map Fields for the Vehicle Object

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

Table 263 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 263. Filterable Fields and User Key Fields on the Vehicle Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Vehicle (parent)	All	VehicleId
		ExternalSystemId
		IntegrationId

Table 263. Filterable Fields and User Key Fields on the Vehicle Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Contact	ContactId	ContactId
	ContactExternalSystemId	ContactExternalSystemId
	ContactIntegrationId	ContactIntegrationId
	ContactFirstName	

Table 264 details the picklists available for the vehicle object.

Table 264. Picklists Available for the Vehicle Object

Field Name
Body
Door
Engine
ExteriorColor
InteriorColor
Location
Make
Model
VehicleOwnedBy
Status
Transmission
Trim
UsedNew
WarrantyType
Year

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

Child Objects (Web Services v1.0)

The following is a list of child objects that are used in Oracle On Demand Web Services. These are objects that are child objects only and are not themselves parent objects:

- "Address" on page 425
- "Attachment" on page 426
- "Audit Trail" on page 426
- "BookUser" on page 426
- "Competitor" on page 426
- "DelegatedUser" on page 427
- "HouseholdTeam" on page 427
- "Interests" on page 428
- "Invitee" on page 428
- "Login History" on page 428
- "Multiple Contact Roles" on page 428
- "OpportunityTeam" on page 428
- "PortfolioTeam" on page 429
- "ProductsDetailed" on page 429
- "PromotionalItemsDropped" on page 429
- "Quota" on page 430
- "Recipient" on page 430
- "Related Account" on page 430
- "Related Contact" on page 430
- "Revenue" on page 430
- "SampleDropped" on page 431
- "SubBook" on page 431
- "Team" on page 431

Address

The address object stores information on the different addresses that are associated with accounts and contacts. It is used to store billing and shipping addresses for accounts. It is also used to store the personal addresses for contacts.

Parent Objects

[Account](#), [Contact](#)

Attachment

The attachment object stores information about a file or URL that is attached to a record in Oracle CRM On Demand.

Parent Objects

[Account](#), [Activity](#), [Campaign](#), [Contact](#), [CustomObject1 - CustomObject3](#), [Lead](#), [Lead](#), [Opportunity](#), [Service Request](#), and [Solution](#).

Audit Trail

The audit trail object stores information about how a service request object is modified from the moment that it is created until a solution for the service request has been found. The audit trail object stores information, such as the created and modified dates for the service request, and also the users who created and updated the service request.

Parent Object

[Service Request](#)

BookUser

The BookUser object stores information about a book user.

For more information on the BookUser fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the book object.

Parent Object

[Book](#)

Competitor

The competitor object stores the information on competitors for your accounts.

Fields

[Table 265](#) details the picklists available for the competitor object.

Table 265. Picklists Available for the Competitor Object

Field Name
RelationshipRole
ReverseRelationshipRole

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

Parent Objects

[Account](#), [Opportunity](#)

DelegatedUser

The DelegatedUser object stores information about a user who can impersonate another user. The delegated user is added so that he or she can access a user's data, for example, when a person is on vacation or leaves the company.

For more information on the DelegatedUser fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the user object.

Parent Object

[User](#)

HouseholdTeam

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

For more information on the household team fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the household object.

Parent Object

[Household](#)

Interests

The Interests object stores information about things in which a contact is interested, such as products, services, or hobbies.

Parent Object

[Contact](#)

Invitee

The Invitee object stores information about invitees to medical education events, including feedback about the invitation.

Parent Object

[MedEd](#)

Login History

The login history object stores information about the currently logged in user, such as the amount of times that the user has logged in, and the dates and times at which the current user logged in.

Parent Object

[Current User](#)

Multiple Contact Roles

The multiple contact roles object stores information on the different roles that a contact can hold within an account. It stores information on the different types of jobs that one contact can hold within your organization. For example, the customer relations manager can also have a role within the sales team to provide valuable feedback to the sales representatives.

Parent Object

[Account](#)

OpportunityTeam

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

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

Parent Object

[Opportunity](#)

PortfolioTeam

The PortfolioTeam object stores information about a team that shares portfolio records

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

Parent Objects

[Account](#), [Product Category](#)

ProductsDetailed

The ProductsDetailed object stores the information on product details for an activity. This is used, for example, to record information about products discussed on sales calls to customers.

For more information on the product detail fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Parent Objects

[Activity](#)

PromotionalItemsDropped

The PromotionalItemsDropped object stores the information on promotional items dropped for an activity.

For more information on the promotional items dropped fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Parent Objects

[Activity](#)

Quota

The quota object stores information about the sales targets of and sales made by the current user.

Parent Object

[Current User](#)

Recipient

The recipient object stores information about a recipient associated with a campaign.

Parent Object

[Campaign](#)

Related Account

The related account object stores information on an account that has a relationship with the parent account in question. The details of the related account child object are inherited from a particular account parent object.

Parent Object

[Account](#)

Related Contact

The related contact object stores information about a contact that has a relationship with the parent contact in question. The details of the related contact child object are inherited from a particular contact parent object.

Parent Object

[Contact](#)

Revenue

The revenue object stores monetary information about accounts, contacts, and their associated opportunities. This includes information on the revenue available, expected revenue, and also information about the products associated with the accounts, contacts, opportunities, and so on.

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

Parent Objects

[Account](#), [Contact](#) and [Opportunity](#)

SampleDropped

The SampleDropped object stores the information on samples for an activity. This is used, for example, to record information about samples left with the customer on sales calls to customers.

For more information on the SampleDropped fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the activity object.

Parent Object

[Activity](#)

SubBook

The Subbook object stores information about a subbook.

For more information on the SubBook fields accessible, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the book object.

Parent Object

[Book](#)

Team

The team object stores information on the team that is assigned to a particular account or contact. In this way, a team of employees can be dedicated to an account or contact, ensuring that the activities, service requests, leads, and opportunities surrounding that account or contact are always kept up-to-date and are attended to regularly.

Parent Objects

[Account](#), [Contact](#), and [CustomObject1 - CustomObject3](#).

Core Parent Objects (Web Services v2.0)

This topic include the standard objects of Oracle CRM On Demand. Objects that are only available with industry-specific editions of the application are covered in the following topics:

■ ["Partner Relationship Management Edition Objects \(Web Services v2.0\)" on page 475](#)

- [“Life Sciences Edition Parent Objects \(Web Services v2.0\)” on page 509](#)
- [“Financial Services Edition Parent Objects \(Web Services v2.0\)” on page 557](#)
- [“Automotive Edition Parent Objects \(Web Services v2.0\)” on page 584](#)

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

- [“Account” on page 433](#)
- [“Activity” on page 435](#)
- [“Address” on page 440](#)
- [“Allotment Usage” on page 442](#)
- [“Asset” on page 443](#)
- [“Book” on page 445](#)
- [“Campaign” on page 447](#)
- [“Category” on page 448](#)
- [“Contact” on page 450](#)
- [“CustomObject” on page 453](#)
- [“Exchange Rate” on page 456](#)
- [“Group” on page 458](#)
- [“Lead” on page 459](#)
- [“Note” on page 461](#)
- [“Opportunity” on page 462](#)
- [“Product” on page 465](#)
- [“Service Allotment” on page 466](#)
- [“Service Request” on page 468](#)
- [“Solution” on page 470](#)
- [“Territory” on page 472](#)
- [“User” on page 474](#)

Objects accessible through the Web Services v2.0 API can reference other objects through a number of reference fields, which are foreign key fields for those other objects. You can determine the objects that are referenced by examining the WSDL file for the referencing object.

NOTE: Information about status keys is not given in this topic because a standard set of status key fields is returned for all parent and child objects accessible through the Web Services v2.0 API. For more information about status keys, see [“Oracle CRM On Demand Status Keys” on page 289](#).

Account

The account object stores information about the companies that you do business with and is also used to track partners and competitors. The methods called on the account object require a list (array) of account objects as an input argument. This list of accounts identifies the records on which the operation is to be carried out.

Adding and Updating Account Addresses

The information in this topic is applicable when the Enable Shared Addresses check box in the company profile is deselected, that is, when addresses are not shared.

NOTE: When Enable Shared Addresses is selected, the address child object, (<ListOfAddress>) cannot be used, and instead the shared address child object (<ListOfSharedAddress>) must be used to associate or dissociate addresses and flag addresses as billing, shipping, or primary addresses. For more information about the shared address child object, see ["Shared Address" on page 634](#) and for more information about the address child object, see ["Address" on page 598](#).

The account object has three types of address:

- 1 Billing address
- 2 Shipping address
- 3 Simple address

The billing and shipping address correspond to fields in the account object. The address child object can correspond to a simple address, but might also correspond to a billing address or a shipping address.

You can use either AccountInsert, AccountUpdate or AccountExecute calls to insert or update billing and shipping addresses for an account. With AccountInsert or AccountUpdate calls, the account parent node with billing and shipping address fields must be present and there must not be any child node. If an address child is present, only a simple address can be added, and not billing and shipping addresses. If the address child or any other child node is not present, only billing and shipping addresses can be added, and not simple addresses.

With the AccountExecute call, both the account parent node and address child node can be present along with billing and shipping address fields at the parent level.

To distinguish whether the address in an address child node is a billing or shipping address:

- If the PrimaryBillToAddressId and AddressId fields have the same value, then the address is a billing address.
- If the PrimaryShipToAddressId and AddressId have the same values, the address is a shipping address.

NOTE: A *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. For more information, see ["Mapping Primary Address Fields Using Web Services" on page 35](#).

Child Objects

[Account Contact](#), [Account Opportunity](#), [Account Team](#), [Activity](#), [Address](#), [Asset](#), [Attachment](#), [Business Plan](#), [Claim](#), [Competitor](#), [Contact](#), [Contact Role](#), [Custom Object 1 through 3](#), [Deal Registration](#), [Financial Account](#), [Financial Account Holder](#), [Financial Plan](#), [Lead](#), [Message Response](#), [Note](#), [Objective](#), [Partner](#), [Policy](#), [Policy Holder](#), [Revenue](#), [Service Request](#), [Shared Address](#), [Special Pricing Product](#).

Methods Called

[Table 266](#) details the methods called by the Account service.

Table 266. Methods Called by Account Service

Method	Name as Defined in Service
"Delete" on page 105	AccountDelete
"Execute" on page 108	AccountExecute
"Insert" on page 112	AccountInsert
"QueryPage" on page 115	AccountQueryPage
"Update" on page 136	AccountUpdate

Fields

[Table 267](#) details the required and read-only fields for the account object.

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

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

[Table 268](#) details the pick map fields for the account object.

Table 268. Pick Map Fields for the Account Object

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

Table 269 provides a list of the user key combinations for the account object.

Table 269. User Key Fields on the Account Object

User Key Field Combinations
Id
ExternalSystemId
AccountName and Location

Table 270 details the picklists available for the account object.

Table 270. Picklists Available for the Account Object

Field Name
AccountType
Priority
Region
CallFrequency
InfluenceType
Route
Status
MarketPotential
MarketSegment

Activity

The activity object stores information on an activity that a user must carry out, for example, a call-back activity for an account. When an activity is created, the user must set the Activity field explicitly to Task or Appointment.

Usage

Oracle On Demand Web Services uses activities to organize, track, and resolve a variety of tasks, from finding and pursuing opportunities to closing service requests. If a task requires multiple steps that one or more people can carry out, activities greatly simplify the job. Activities can help to:

- Define and assign the task
- Provide information to complete the task
- Track the progress of the task

- Track costs and bill for the task

NOTE: The Owner field is not accessible for the activity object in Web Services v2.0.

Support for Recurring Events

The Activity service allows access to the iCRMId field. This field is used to support the synchronization of recurring appointments between Oracle CRM On Demand Desktop and Oracle CRM On Demand.

Microsoft Outlook supports a variety of recurring appointments (for example, meetings) where an appointment can occur daily, weekly, monthly, and so on. In addition, Outlook also supports exceptions within a recurring series of appointments, for example, a meeting that occurs at the same time each day in the week apart from one day where the meeting is at a different time. However, Oracle CRM On Demand does not support recurring activities, so each occurrence of a recurring appointment is created as an individual activity in Oracle CRM On Demand.

The iCRMId field contains a representation of the recurrence parameters and is required to reassemble the recurring appointment and its exceptions within Outlook if there are any changes to the individual activities in Oracle CRM On Demand. The field is 450 characters in length, is indexed, and contains:

- The unique Id of the recurring appointment generated by Outlook.
- The recurrence rule (RRULE), for example:
FREQ=MONTHLY; UNTIL=20110207T144325; INTERVAL=2; BYDAY=MO, TU, WE, TH, FR; BYSETPOS=-1
- VTIMEZONE, the timezone specified for each Outlook appointment, for example:
BEGIN: VTIMEZONE
TZID: (UTC+02:00)
BEGIN: STANDARD
DTSTART: 16011028T040000
TZOFFSETFROM: +0300
TZOFFSETTO: +0200
END: STANDARD
BEGIN: DAYLIGHT
DTSTART: 16010325T030000
TZOFFSETFROM: +0200
TZOFFSETTO: +0300
END: DAYLIGHT
END: VTIMEZONE
- The offset of the occurrence from the start of the series (int).

- The hash value of the original series fields (int). This includes the hash value of location, appt startdate, appt enddate, subject, and description.

For more information about the synchronization of recurring appointments between Oracle CRM On Demand Desktop and Oracle CRM On Demand, see *Oracle CRM On Demand Desktop Administration Guide*.

Child Objects

[Address](#), [Sales Assessment Value](#), [Attachment](#), [Attendee Call](#), [Contact](#), [Custom Object 1 through 3](#), [Message Response](#), [Products Detailed](#), [Promotional Items Dropped](#), [Sample Dropped](#), [Sample Transaction](#), [Signature](#), [User](#).

Methods Called

[Table 271](#) details the methods called by the Activity service.

Table 271. Methods Called by Activity Service

Method	Name as Defined in Service
"Delete" on page 105	ActivityDelete
"Execute" on page 108	ActivityExecute
"Insert" on page 112	ActivityInsert
"QueryPage" on page 115	ActivityQueryPage
"Update" on page 136	ActivityUpdate

Fields

[Table 272](#) details the required and read-only fields for the activity object.

Table 272. 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

Table 272. Required and Read-Only Fields for the Activity Object

Field Name	Type
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

[Table 273](#) details the pick map fields for the activity object.

Table 273. Pick Map Fields for the Activity Object

Pick Map Field	Maps To
OwnerAlias	OwnerId
AccountName	AccountId
AccountLocation	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
CampaignExternalSystemId	CampaignId
CampaignIntegrationId	CampaignId
CampaignName	CampaignId
LeadExternalSystemId	LeadId
LeadIntegrationId	LeadId
MedEdEventExternalSystemId	MedEdEventId
MedEdEventIntegrationId	MedEdEventId
OpportunityExternalSystemId	OpportunityId
OpportunityIntegrationId	OpportunityId
PortfolioExternalSystemId	PortfolioId

Table 273. Pick Map Fields for the Activity Object

Pick Map Field	Maps To
PortfolioIntegrationId	PortfolioId
ServiceRequestNumber	ServiceRequestId
ServiceRequestExternalSystemId	ServiceRequestId
ServiceRequestIntegrationId	ServiceRequestId
FundRequestExternalSystemId	FundRequestId
FundRequestIntegrationId	FundRequestId

Table 274 provides a list of the user key combinations for the activity object.

Table 274. User Key Fields on the Activity Object

User Key Field Combinations
Description
Id
ExternalSystemId

Table 275 details the picklists available for the activity object.

Table 275. Picklists Available for the Activity Object

Field Name
AccountName
AccountLocation
AccountIntegrationId
AccountExternalSystemId
OpportunityName
ServiceRequestNumber
ServiceRequestIntegrationId
ServiceRequestExternalSystemId
DelegatedByExternalSystemId
PrimaryContactIntegrationId
PrimaryContactExternalSystemId

Table 275. Picklists Available for the Activity Object

Field Name
MedEdEventIntegrationId
MedEdEventExternalSystemId
FundRequestExternalId
LeadIntegrationId
LeadExternalSystemId
CampaignIntegrationId
CampaignExternalSystemId
ActivitySubtype
ResolutionCode
PublishInternal
Status
Issue
Indication

Address

The shared address object stores information about addresses that can be shared between account, contact, dealer, and partner records.

Usage

This object is applicable only for companies that have the Enable Shared Addresses check box selected on the company profile. In this case, addresses can be associated with multiple account, contact, dealer, or partner records.

After a shared address object is inserted, it can be shared between accounts, contacts, dealers and partners by inserting a child shared address object for those record types. When a child shared address is inserted, it can be flagged as a billing address or shipping address for accounts, dealers, and partners, or as a primary contact address for contacts. When this occurs, the billing, shipping, or primary contact fields in the parent object are populated from the shared address fields. For more information about the shared address child object, see [“Shared Address” on page 634](#).

NOTE: If the Validate Shared Addresses option is selected for the company, an address can only be shared after it is validated.

When a shared address is updated through the Address service, all shared address child objects for the address are updated. Also, the address fields in the associated account, contact, dealer, or partner objects are updated, if the address deleted was flagged as a billing, shipping, or primary contact address.

To delete a shared address, the Manage Addresses privilege is required. When a shared address is deleted through the Address service, all shared address child objects for the address are dissociated from their parent objects. Also, the address fields in the associated account, contact, dealer, or partner objects are deleted, if the address deleted was flagged as a billing, shipping, or primary contact address. For more information about the shared address functionality, see *Oracle CRM On Demand Online Help*.

Methods Called

Table 276 details the methods called by the Address service.

Table 276. Methods Called by Address Service

Method	Name as Defined in Service
“Delete” on page 105	SharedAddressDelete
“Execute” on page 108	SharedAddressExecute
“Insert” on page 112	SharedAddressInsert
“QueryPage” on page 115	SharedAddressQueryPage
“Update” on page 136	SharedAddressUpdate

Fields

Table 277 details the required and read-only fields for the shared address object.

Table 277. Required and Read-Only Fields for the Shared Address Object

Field Name	Type
AddressName	Required
Audit Fields	Read-only

Pickmaps

Table 278 details the pick map fields for the shared address object.

Table 278. Pick Maps Fields for the Shared Address Object

Pick Map Field	Maps To
CustomObjectnName	CustomObjectnId
CustomObjectnExternalSystemId	CustomObjectnId

Table 280 details the user keys for the shared address object.

Table 279. User Keys for the Shared Address Object

Field Name
Id
ExternalSystemId
AddressName

Table 280 details the picklists available for the shared address object.

Table 280. Picklists for the Shared Address Object

Field Name
CurrencyCode

Allotment Usage

The allotment usage object stores information about the historical allotment usage for a company.

Usage

The Allotment Usage service is only accessible to users with the Manage Company privilege.

There are various elements of the ListOfAllotmentUsage argument that specify the service allotment usage data:

- **AggregationType**. The aggregation type.
- **AggregationPeriodTotalUsage**. The total usage in the aggregation period.
- **AggregationPeriodTimestamp**. The aggregation period.
- **AllotmentLimitId**. The value set for the allotment by Customer Care.
- **AllotmentUnit**. The allotment unit of measure.
- **AllotmentType**. The allotment type.
- **AllotmentLimitName**. The allotment name.
- **AllotmentLimitDisplayName**. The allotment name in the language currently active for the company.

There are also various audit fields returned.

For more information about administration of service allotments, see *Oracle CRM On Demand Online Help*.

Methods Called

[Table 281](#) details the methods called by the Allotment Usage service.

Table 281. Methods Called by Allotment Usage Service

Method	Name as Defined in Service
"QueryPage" on page 115	AllotmentUsageQueryPage

Fields

[Table 282](#) details the required and read-only fields for the allotment usage object.

Table 282. Required and Read-Only Fields for the Allotment Usage Object

Field Name	Type
AllotmentLimitName	Required
Audit Fields	Read-only

[Table 283](#) details the user keys for the allotment usage object.

Table 283. User Keys for the Allotment Usage Object

Field Name
AllotmentLimitName

Asset

The asset object stores information on the assets held by your accounts, for example, the products that an account has purchased.

Child Objects

None

Usage

Oracle On Demand Web Services uses assets to manage products through their life cycle. It is also used by your accounts to register products, receive product news and literature, track warranty agreements, and receive recommendations on scheduled services.

Methods Called

Table 284 details the methods called by the Asset service.

Table 284. Methods Called by Asset Service

Method	Name as Defined in Service
"Delete" on page 105	AssetDelete
"Execute" on page 108	AssetExecute
"Insert" on page 112	AssetInsert
"QueryPage" on page 115	AssetQueryPage
"Update" on page 136	AssetUpdate

Fields

Table 285 details the required and read-only fields for the asset object.

Table 285. 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 286 details the pick map fields for the asset object.

Table 286. 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 287 provides a list of the user key combinations for the asset object.

Table 287. User Key Fields on the Asset Object

User Key Field Combinations
Id
ExternalSystemId
ProductId
ProductName
ProductExternalId

Table 288 details the picklists available for the asset object.

Table 288. Picklists Available for the Asset Object

Field Name
Warranty
Contract

Book

The book object provides a way of segmenting data according to the organizational units of your business, such as territories or products. Administrators can create book hierarchies based on how they want to organize your information, and then set up users to have the appropriate level of access to books.

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

Methods Called

Table 289 details the methods called by the Book service.

Table 289. Methods Called by Book Service

Method	Name as Defined in Service
"Delete" on page 105	BookDelete
"Execute" on page 108	BookExecute

Table 289. Methods Called by Book Service

Method	Name as Defined in Service
"Insert" on page 112	BookInsert
"QueryPage" on page 115	BookQueryPage
"Update" on page 136	BookUpdate

Fields

[Table 290](#) details the required and read-only fields for the book object.

Table 290. Required and Read-Only Fields for the Book Object

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

[Table 291](#) details the pick map fields for the book object.

Table 291. Pick Map Fields for the Book Object

Pick Map Field	Maps To
BookName	BookId

[Table 292](#) provides a list of the user key combinations for the book object.

Table 292. User Key Fields on the Book Object

User Key Field Combinations
Id
BookName

[Table 293](#) details the picklists available for the book object.

Table 293. Picklists Available for the Book Object

Field Name
BookType
ParentBookName

Campaign

The campaign object provides a mechanism for marketing products and services to customers and prospects. The campaign object is the primary way in which new products and services are marketed to customers and prospects.

Child Objects

[Activity](#), [Application](#), [Attachment](#), [Book](#), [Financial Account](#), [Financial Plan](#), [Lead](#), [MDF Request](#), [MedEd](#), [Note](#), [Policy](#)

Methods Called

[Table 294](#) details the methods called by the Campaign service.

Table 294. Methods Called by Campaign Service

Method	Name as Defined in Service
"Delete" on page 105	CampaignDelete
"Execute" on page 108	CampaignExecute
"Insert" on page 112	CampaignInsert
"QueryPage" on page 115	CampaignQueryPage
"Update" on page 136	CampaignUpdate

Fields

[Table 295](#) details the required and read-only fields for the campaign object.

Table 295. 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 296](#) details the pick map field for the campaign object.

Table 296. Pick Map Field for the Campaign Object

Pick Map Field	Maps To
Owner	OwnerId

[Table 297](#) provides a list of the user key combinations for the campaign object.

Table 297. User Key Fields on the Campaign Object

User Key Field Combinations
Id
ExternalSystemId
SourceCode
CampaignName

[Table 298](#) details the picklists available for the campaign object.

Table 298. Picklists Available for the Campaign Object

Field Name
CampaignType
Status

Related Topic

[Current User](#) and [Opportunity](#)

Category

The category object allows you to logically sort products into groups, where each product is in some way related to the other products in the category.

The category object is equivalent to the Web Services V1.0 product category object.

Methods Called

[Table 299](#) details the methods called by the Category service.

Table 299. Methods Called by Category Service

Method	Name as Defined in Service
"Delete" on page 105	CategoryDelete
"Execute" on page 108	CategoryExecute
"Insert" on page 112	CategoryInsert
"QueryPage" on page 115	CategoryQueryPage
"Update" on page 136	CategoryUpdate

Fields

[Table 300](#) details the required and read-only fields for the category object.

Table 300. Required and Read-Only Fields for the Category Object

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

[Table 301](#) details the pick map field for the category object.

Table 301. Pick Map Field for the Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

[Table 302](#) details the user keys for the category object.

Table 302. User Keys for the Category Object

Field Name
Id

Table 302. User Keys for the Category Object

Field Name
ExternaSystemId
Name

Contact

The contact object stores information on individuals with whom your organization has a relationship. It allows the user to store information on individuals who are external to your company, but who are associated with the business process. Contacts stored in the Oracle CRM On Demand database can also be associated with an account.

Adding and Updating Contact Addresses

The information in this topic is applicable when the Enable Shared Addresses check box in the company profit is deselected, that is, when addresses are not shared.

NOTE: When Enable Shared Addresses is selected, the address child object, (<ListOfAddress>) cannot be used, and instead the shared address child object (<ListOfSharedAddress>) must be used to associate or dissociate addresses and flag addresses as billing, shipping, or primary addresses. For more information about the shared address child object, see [“Shared Address” on page 634](#) and for more information about the address child object, see [“Address” on page 598](#).

The contact object has three types of address:

- 1 Primary address
- 2 Alternate address
- 3 Simple address

If the contact is associated with an account, the account’s billing address becomes the contact’s primary address, that is, the account’s billing address fields populate the contact’s primary address fields. The primary address fields are read-only in the contact record. The primary address might also be called the account address.

The alternate address is simply the contact’s address. Alternate address fields at the parent level are editable and this address is added to address children with the Primary checkbox in the UI selected. The alternate address might also be called the contact address.

You can use either ContactInsert, ContactUpdate or ContactExecute calls to insert or update primary or alternate addresses for a contact. With ContactInsert or ContactUpdate calls, the contact parent node with primary and alternate address fields must be present, and there must not be any child node. If an address child is present, only a simple address can be added, and not primary and alternate addresses. If the address child or any other child node is not present, only primary and alternate addresses can be added, and not simple addresses.

With the ContactExecute call, both the contact parent node and address child node can be present along with primary and alternate address fields at the parent level.

NOTE: A *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. For more information, see [“Mapping Primary Address Fields Using Web Services” on page 35](#).

Child Objects

Account, Activity, Address, Attachment, Business Plan, Blocked Product, Campaign, Claim, Contact Best Time, Contact State License, Coverage, Custom Object 1 through 3, Deal Registration, Financial Account, Financial Account Holder, Financial Plan, Interests, Involved Party, Lead, Message Response, Note, Objective, Opportunity, Plan Contact, Policy, Policy Holder, Related Contact, Revenue, Service Request, Signature, Shared Address, Special Pricing Product, Contact Team.

Methods Called

[Table 303](#) details the methods called by the Contact service.

Table 303. Methods Called by Contact Service

Method	Name as Defined in Service
“Delete” on page 105	ContactDelete
“Execute” on page 108	ContactExecute
“Insert” on page 112	ContactInsert
“QueryPage” on page 115	ContactQueryPage
“Update” on page 136	ContactUpdate

Fields

[Table 304](#) details the required and read-only fields for the contact object.

Table 304. 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

Table 304. Required and Read-Only Fields for the Contact Object

Field Name	Type
Manager	Read-only
PrimaryAddressId	Read-only

Table 305 details the pick map fields for the contact object.

Table 305. Pick Map Fields for the Contact Object

Pick Map Field	Maps To
AccountName	AccountId
Owner	AssignedToAlias
SourceCampaignName	SourceCampaignId
ManagerExternalSystemId	ManagerId

Table 306 provides a list of the user key combinations for the contact object.

Table 306. User Key Fields on the Contact Object

User Key Field Combinations
ContactFirstName and ContactLastName
Id
ExternalSystemId

Table 307 details the picklists available for the contact object.

Table 307. Picklists Available for the Contact Object

Field Name
ContactType
LeadSource
MrMrs
BestTimeToCall
BlockedReasonCode
CallFrequency

Table 307. Picklists Available for the Contact Object

Field Name
CurrentInvestmentMix
Degree
ExperienceLevel
Gender
InvestmentHorizon
LifeEvent
MaritalStatus
MarketPotential
Objective
OwnOrRent
PrimaryGoal
RiskProfile
Route
Segment
Tier

[Table 308](#) details a number of contact object fields that you must not use for customer integrations.

Table 308. Contact Object Fields That You Must Not Use

Field Name
PartyTypeCode
PartyUID
PersonUID

CustomObject

The CustomObject services expose the functionality of the CustomObject objects to external applications.

TIP: The reference information for each of Custom Objects follows the same pattern. In this topic, the information for CustomObject1 is given as an example.

The name of some CustomObject fields in the WSDL files for Web Services v2.0 differ from the names used in the WSDL files for Web Services v1.0. For example, the Web Services v1.0 fields ParentExternalSystemId and ParentName correspond to CustomObject1ExternalSystemId and CustomObject1Name for Web Services v2.0.

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

Child Objects

[Attachment](#).

Methods Called

[Table 309](#) details the methods called by the CustomObject1 service. The methods for the other Custom Objects follow the same pattern.

Table 309. Methods Called by CustomObject1 Service

Method	Name as Defined in Service
"Delete" on page 105	CustomObject1Delete
"Execute" on page 108	CustomObject1Execute
"Insert" on page 112	CustomObject1Insert
"QueryPage" on page 115	CustomObject1QueryPage
"Update" on page 136	CustomObject1Update

Fields

[Table 310](#) details the required and read-only fields for the CustomObject1 object.

Table 310. Required and Read-Only Fields for the CustomObject1 Object

Field Name	Type
Id	Required
CustomObject1ExternalSystemID	Required
CustomObject1IntegrationId	Required

Table 311 details the pick map fields for the CustomObject1 object.

Table 311. 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	CustomObject n Id
CustomObject n IntegrationId	CustomObject n Id
CustomObject n Name	CustomObject n Id
DealerName	DealerId
HouseholdExternalId	HouseholdId
HouseholdIntegrationId	HouseholdId
HouseholdName	HouseholdId
LeadExternalId	LeadId
LeadFirstName	LeadId
LeadFullName	LeadId
LeadIntegrationId	LeadId
LeadLastName	LeadId
Owner	OwnerId
OpportunityExternalId	OpportunityId
OpportunityIntegrationId	OpportunityId

Table 311. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
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 312](#) provides a list of the user key combinations for the CustomObject1 object.

Table 312. User Key Fields on the CustomObject1 Object

User Key Field Combinations
Id
ExternalSystemId

Exchange Rate

The Exchange rate object stores information about exchange rates for the currencies used by a company.

Usage

Administrators can create new exchange rates, update, or delete existing exchange rates, and query the exchange rate history for their company.

All fields are required when inserting new exchange rate records. For insert operations, the ToCurrencyCode value must be the company's default currency value.

Only the FromCurrencyCode, ToCurrencyCode, and ExchangeDate fields are required for update delete operations. However, these fields are not editable for an update operation.

The date value in the ExchangeDate field must be specified in the format: *yyyy-mm-dd*. For example, a value of 2011-10-12 specifies the date as 12th October 2011.

The ExchangeRateHistory child object can be queried to give the history of exchange rates for the company.

Methods Called

Table 313 details the methods called by the Exchange Rate service.

Table 313. Methods Called by Exchange Rate Service

Method	Name as Defined in Service
"Delete" on page 105	ExchangeRateDelete
"Execute" on page 108	ExchangeRateExecute
"Insert" on page 112	ExchangeRateInsert
"QueryPage" on page 115	ExchangeRateQueryPage
"Update" on page 136	ExchangeRateUpdate

Fields

Table 314 details the required and read-only fields for the exchange rate object.

Table 314. Required and Read-Only Fields for the Exchange Rate Object

Field Name	Type
FromCurrencyCode	Required
ToCurrencyCode	Required
ExchangeDate	Required
ExchangeRate	Required

Table 315 provides a list of the user key combinations for the exchange rate object.

Table 315. User Key Fields on the Exchange Rate Object

User Key Field Combinations
FromCurrencyCode
ToCurrencyCode

Table 315. User Key Fields on the Exchange Rate Object

User Key Field Combinations
ExchangeDate

Group

The Group object allows you to create groups to which users can be added. Users can only be a member of one group, and groups can contain many users.

The group object is equivalent to the Web Services V1.0 user group object.

Methods Called

Table 316 details the methods called by the Group service.

Table 316. Methods Called by Group Service

Method	Name as Defined in Service
"Delete" on page 105	GroupDelete
"Execute" on page 108	GroupExecute
"Insert" on page 112	GroupInsert
"QueryPage" on page 115	GroupQueryPage
"Update" on page 136	GroupUpdate

Fields

Table 317 details the required and read-only fields for the group object.

Table 317. Required and Read-Only Fields for the Group Object

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

Table 318 details the pick map field for the group object.

Table 318. Pick Map Field for the Group Object

Pick Map Field	Maps To
UserIntegrationId	UserId
UserExternalSystemId	UserId

Table 319 provides a list of the user key combinations for the group object.

Table 319. User Key Fields on the Group Object

User Key Field Combinations
Id
ExternalSystemId
Name

Lead

The lead object stores information on a company or individual with whom an opportunity can be created. It allows the user to identify the companies that might be interested in a product or service. Leads are usually generated as part of a marketing campaign.

NOTE: A *PrimaryAddressLine1* field is used to dynamically map the primary address field from an external application to the primary address field in Oracle CRM On Demand. For more information, see [“Mapping Primary Address Fields Using Web Services” on page 35](#).

Child Objects

[Activity](#), [Attachment](#), [Custom Object 1 through 3](#).

Methods Called

Table 320 details the methods called by the Lead service.

Table 320. Methods Called by Lead Service

Method	Name as Defined in Service
“Delete” on page 105	LeadDelete
“Execute” on page 108	LeadExecute

Table 320. Methods Called by Lead Service

Method	Name as Defined in Service
"Insert" on page 112	LeadInsert
"QueryPage" on page 115	LeadQueryPage
"Update" on page 136	LeadUpdate

Fields

[Table 321](#) details the required and read-only fields for the lead object.

Table 321. 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 322](#) details the pick map fields for the lead object.

Table 322. 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 323 provides a list of the user key combinations of the lead object.

Table 323. User Key Fields on the Lead Object

User Key Field Combinations
Id
ExternalSystemId
LeadFirstName and LeadLastName

Table 324 details the picklists available for the lead object.

Table 324. Picklists Available for the Lead Object

Field Name
Country
MrMrs
Rating
Source
StateProvince
Status

Note

The note object stores information about the notes available in the Message Center in the Oracle CRM On Demand application. The notes can be sent from users or can store extra information (as a note) on a parent object. This allows employees who are working on a particular record to add extra information as they see fit. For example, when talking to a contact, an employee might notice that the contact is not happy with a service provided. The employee can record this information in a note so that any other employees who talk to the contact are aware of the contact's dissatisfaction.

Methods Called

Table 325 details the methods called by the Note service.

Table 325. Methods Called by Note Service

Method	Name as Defined in Service
"Delete" on page 105	NoteDelete
"Execute" on page 108	NoteExecute

Table 325. Methods Called by Note Service

Method	Name as Defined in Service
"Insert" on page 112	NoteInsert
"QueryPage" on page 115	NoteQueryPage
"Update" on page 136	NoteUpdate

Fields

[Table 326](#) details the required and read-only fields for the note object.

Table 326. Required and Read-Only Fields for the Note Object

Field Name	Type
Subject	Required
NotId	Read-only
OwnerId	Read-only
OwnerAlias	Read-only
ParentNotId	Read-only
SourceId	Read-only
SourceName	Read-only
Audit Fields	Read-only

[Table 327](#) provides a list of the user key combinations for the note object.

Table 327. User Key Fields on the Note Object

User Key Field Combinations
NotId

Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Child Objects

Activity, Attachment, Competitor, Contact, Contact Role, Custom Object 1 through 3, Deal Registration, Lead, Note, Opportunity Child, Opportunity Team, Partner, Plan Opportunity, Revenue, Special Pricing Product.

Methods Called

Table 328 details the methods called by the Opportunity service.

Table 328. Methods Called by Opportunity Service

Method	Name as Defined in Service
"Delete" on page 105	OpportunityDelete
"Execute" on page 108	OpportunityExecute
"Insert" on page 112	OpportunityInsert
"QueryPage" on page 115	OpportunityQueryPage
"Update" on page 136	OpportunityUpdate

Fields

Table 329 details the required and read-only fields for the opportunity object.

Table 329. 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 330 details the pick map fields for the opportunity object.

Table 330. Pick Map Fields for the Opportunity Object

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

Table 331 provides a list of the user key combinations for the opportunity object.

Table 331. User Key Fields on the Opportunity Object

User Key Field Combinations
Id
ExternalSystemId
Name

Table 332 details the picklists available for the opportunity object.

Table 332. Picklists Available for the Opportunity Object

Field Name
LeadSource
Priority
Probability
ReasonWonLost
Status
Type
Year
Make
Model

Product

The product object allows you to define and record details about a product or service that your company sells to its customers, including information on product price, category, and so on.

Child Objects

[Price List Line Item](#)

Methods Called

[Table 333](#) details the methods called by the Product service.

Table 333. Methods Called by Product Service

Method	Name as Defined in Service
"Execute" on page 108	ProductExecute
"Insert" on page 112	ProductInsert
"QueryPage" on page 115	ProductQueryPage
"Update" on page 136	ProductUpdate

Fields

[Table 334](#) details the required and read-only fields for the product object.

Table 334. Required and Read-Only Fields for the Product Object

Field Name	Type
ProductName	Required
Audit Fields	Read-only

[Table 335](#) details the pick map field for the product object.

Table 335. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 336 details the user keys for the product object.

Table 336. User Keys for the Product Object

Field Name
ProductName
Id
ExternalSystemId

Table 337 details the picklists available for the product object.

h

Table 337. Picklists Available for the Product Object

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

Service Allotment

The service allotment object stores information about the service allotments for a company.

Usage

The Service Allotment service is only accessible to users with the Manage Company privilege.

There is only one method, `ServiceAllotmentQueryPage`, which returns the service allotment data for the company.

There are various elements of the `ListOfServiceAllotment` argument that specify the service allotment data:

- **Name**. The allotment name.
- **Value**. The value set for the allotment by Customer Care
- **CurrentUsage**. The amount of the allotment currently used.
- **AlertThreshold**. The threshold value at which a warning email is sent.
- **AlertEnabled**. Whether an alert is enabled for the allotment.
- **EmailAlertList**. The email addresses of the users who will receive a warning email.
- **AllotmentEmailSentDate**. The date when an email notifying full usage of the allotment was sent.
- **ThresholdEmailSentDate**. The date when a warning email for the allotment was sent.
- **DisplayName**. The allotment name in the language currently active for the company.
- **AllotmentUnit**. The allotment unit of measure.
- **AllotmentType**. The type of allotment.
- **RemainingUsage**. The amount of the allotment remaining.
- **Description**. A description of the allotment.
- **ListOfAllotmentUsage**. The allotment usage data:
 - **AggregationType**. The aggregation type.
 - **AggregationPeriodTotalUsage**. The total usage in the aggregation period.
 - **AggregationPeriodTimestamp**. The aggregation period.
 - **AllotmentLimitId**. The value set for the allotment by Customer Care.
 - **AllotmentUnit**. The allotment unit of measure.
 - **AllotmentType**. The allotment type.
 - **AllotmentLimitName**. The allotment name.
 - **AllotmentLimitDisplayName**. The allotment name in the language currently active for the company.

There are also various audit fields returned.

For more information about administration of service allotments, see *Oracle CRM On Demand Online Help*.

Methods Called

[Table 338](#) details the methods called by the Service Allotment service.

Table 338. Methods Called by Service Allotment Service

Method	Name as Defined in Service
"QueryPage" on page 115	ServiceAllotmentQueryPage

Fields

[Table 339](#) details the required and read-only fields for the service allotment object.

Table 339. Required and Read-Only Fields for the Service Allotment Object

Field Name	Type
Name	Required
Audit Fields	Read-only

[Table 340](#) details the user keys for the service allotment object.

Table 340. User Keys for the Service Allotment Object

Field Name
Name

Service Request

The service request object allows customers to request information or assistance with a problem related to products or services purchased from your company. Service requests can be ranked for severity and prioritized accordingly.

Child Objects

[Attachment](#), [Note](#).

Methods Called

Table 341 details the methods called by the Service request service.

Table 341. Methods Called by Service Request Service

Method	Name as Defined in Service
"Delete" on page 105	ServiceRequestDelete
"Execute" on page 108	ServiceRequestExecute
"Insert" on page 112	ServiceRequestInsert
"QueryPage" on page 115	ServiceRequestQueryPage
"Update" on page 136	ServiceRequestUpdate

Fields

Table 342 details the required and read-only fields for the service request object.

Table 342. Required and Read-Only Fields for the Service Request Object

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

Table 343 details the pick map fields for the service request object.

Table 343. Pick Map Fields for the Service Request Object

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

Table 344 provides a list of the user key combinations for the service request object.

Table 344. User Key Fields on the Service Request Object

User Key Field Combinations
Id
ExternalSystemId
SRNumber

Table 345 details the picklists available for the service request object.

Table 345. Picklists Available for the Service Request Object

Field Name
Area
Cause
Priority
Source
Status
Type

Solution

The solution object stores information on solutions to customer problems or service requests. Solutions can be reused if the same problem is identified with a product or service. This prevents the duplication of work for customer service representatives.

Child Object

[Attachment](#), [Book](#), [Message Response](#), [Messaging Plan](#), [Messaging Plan Item](#), [Messaging Plan Item Relation](#).

Methods Called

Table 346 details the methods called by the Solution service.

Table 346. Methods Called by Solution Service

Method	Name as Defined in Service
"Delete" on page 105	SolutionDelete
"Execute" on page 108	SolutionExecute
"Insert" on page 112	SolutionInsert
"QueryPage" on page 115	SolutionQueryPage
"Update" on page 136	SolutionUpdate

Fields

Table 347 details the required and read-only fields for the solution object.

Table 347. 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 348 provides a list of the user key combinations for the solution object.

Table 348. User Key Fields on the Solution Object

User Key Field Combinations
Name
Id
ExternalSystemId

Table 349 details the picklists available for the solution object.

Table 349. Picklists Available for the Solution Object

Field Name
Area
Cause
Priority
Source
Status
Type

Territory

The territory object allows you to store information about the sales territory that is assigned to a user. This information includes the territory name, a description, the currency code, and the sales quota for the territory.

Methods Called

Table 350 details the methods called by the Territory service.

Table 350. Methods Called by Territory Service

Method	Name as Defined in Service
"Delete" on page 105	TerritoryDelete
"Execute" on page 108	TerritoryExecute
"Insert" on page 112	TerritoryInsert
"QueryPage" on page 115	TerritoryQueryPage
"Update" on page 136	TerritoryUpdate

Fields

[Table 351](#) details the required and read-only fields for the territory object.

Table 351. Required and Read-Only Fields for the Territory Object

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

[Table 352](#) details the pick map field for the territory object.

Table 352. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

[Table 353](#) details the user key for the territory object.

Table 353. User Key for the Territory Object

Field Name
Id
ExternalSystemId
TerritoryName

[Table 354](#) details the picklist fields available for the territory object.

Table 354. Picklists Available for the Territory Object

Field Name
ParentTerritoryExternalSystemId
ParentTerritoryIntegrationId

User

The user object allows you to define and record details of all users in Oracle CRM On Demand, for example, name, position, contact details, manager, and so on. It enables queries to be run on all users, and enables an administrator to insert and update a user's profile.

Usage

The UserLoginId and UserSignInId fields must be used as follows:

- **UserLoginId.** Used for creating user records through the User Web service.
- **UserSignInId.** Used as the user name for logging in and authenticating using Web services. Also, used for queries, as using UserLoginId is not allowed for queries.

Child Object

[Delegated User](#)

Methods Called

[Table 355](#) details the methods called by the User service.

Table 355. Methods Called by User Service

Method	Name as Defined in Service
"Execute" on page 108	UserExecute
"Insert" on page 112	UserInsert
"QueryPage" on page 115	UserQueryPage
"Update" on page 136	UserUpdate

Fields

[Table 356](#) details the required and read-only fields for the user object.

Table 356. Required and Read-Only Fields for the User Object

Field Name	Type
FirstName	Required
LastName	Required
UserLoginId	Required
UserSignInId	Required
Alias	Required

Table 356. Required and Read-Only Fields for the User Object

Field Name	Type
EmailAddr	Required
Role	Required
Status	Required
Audit Fields	Read-only
LastSignInDateTime	Read-only
ManagerFullName	Read-only

[Table 357](#) details the pick map field for the user object.

Table 357. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

[Table 358](#) provides a list of user key combinations for the user object.

Table 358. User Key Fields on the User Object

User Key Field Combinations
Id
ExternalSystemId
Alias

Partner Relationship Management Edition Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Partner Relationship Management Edition. For these parent objects, all child objects are also accessible.

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

The following objects are detailed in this topic:

■ [“Accreditation” on page 476](#)

- ["Accreditation Request" on page 478](#)
- ["Application" on page 480](#)
- ["Certification" on page 482](#)
- ["Certification Request" on page 484](#)
- ["Course" on page 486](#)
- ["Course Enrollment" on page 488](#)
- ["Deal Registration" on page 490](#)
- ["Exam" on page 492](#)
- ["Exam Registration" on page 494](#)
- ["Fund" on page 496](#)
- ["MDF Request" on page 498](#)
- ["Partner" on page 500](#)
- ["Price List" on page 502](#)
- ["Price List Line Item" on page 503](#)
- ["Special Pricing Product" on page 505](#)
- ["Special Pricing Request" on page 506](#)

Accreditation

The accreditation object stores information about a partner company's accreditation for selling products, providing services, or both. Accreditations are an acknowledgement of a partner company's skill, competence, and training in selling a particular line of products or offering services such as design, installation, upgrade, and so on. A partner company is awarded an accreditation upon satisfying several requirements; one of which is usually a minimum number (or percentage) of employees with specific certifications. In Oracle CRM On Demand, an accreditation can be held only by a partner company. Individuals cannot hold accreditations; instead, they hold certifications.

Child Objects

[Accreditation](#), [Accreditation Request](#), [Attachment](#), [Book](#), [Category](#), [Certification](#), Custom Object 01-03, Custom Object 4 through 15, [Product](#), [Solution](#).

Methods Called

Table 359 details the methods called by the Accreditation service.

Table 359. Methods Called by Accreditation Service

Method	Name as Defined in Service
"Delete" on page 105	AccreditationDelete
"Execute" on page 108	AccreditationExecute
"Insert" on page 112	AccreditationInsert
"QueryPage" on page 115	AccreditationQueryPage
"Update" on page 136	AccreditationUpdate

Fields

Table 360 details the required and read-only fields for the accreditation object.

Table 360. Required and Read-Only Fields for the Accreditation Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

Table 361 details the pick map fields for the accreditation object.

Table 361. Pick Map Fields for the Accreditation Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
PrimaryProductCategoryName	PrimaryProductCategoryId

Table 361. Pick Map Fields for the Accreditation Object

Pick Map Field	Maps To
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId

[Table 362](#) provides a list of the user key combinations for the accreditation object.

Table 362. User Key Fields on the Accreditation Object

User Key Field Combinations
Id
Name

[Table 363](#) details the picklists available for the accreditation object.

Table 363. Picklists Available for the Accreditation Object

Field Name
PaymentOption
Status
Type

Accreditation Request

The accreditation request object stores information about a partner company's request for accreditation.

Child Objects

None

Methods Called

Table 364 details the methods called by the Accreditation Request service.

Table 364. Methods Called by Accreditation Request Service

Method	Name as Defined in Service
“Delete” on page 105	AccreditationRequestDelete
“Execute” on page 108	AccreditationRequestExecute
“Insert” on page 112	AccreditationRequestInsert
“QueryPage” on page 115	AccreditationRequestQueryPage
“Update” on page 136	AccreditationRequestUpdate

Fields

Table 365 details the required and read-only fields for the accreditation request object.

Table 365. Required and Read-Only Fields for the Accreditation Request Object

Field Name	Type
AccreditationId	Required
PartnerId	Required
OwnerId	Required
Audit Fields	Read-only

Table 366 details the pick map fields for the accreditation request object.

Table 366. Pick Map Fields for the Accreditation Request Object

Pick Map Field	Maps To
AccreditationName, AccreditationNum, AccreditationStatus, AccreditationType, AccreditationExternalSystemId	AccreditationId
PartnerChannelAccountManagerAlias, PartnerIntegrationId, PartnerExternalSystemId, PartnerLocation, PartnerMainPhone, PartnerName	PartnerId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
PlanName, PlanType, PlanStatus, PlanDescription, PlanExternalSystemId	PlanId

[Table 367](#) provides a list of the user key combinations for the accreditation request object.

Table 367. User Key Fields on the Accreditation Request Object

User Key Field Combinations
Id
AccreditationId, PartnerId and RequestDate

[Table 368](#) details the picklists available for the accreditation request object.

Table 368. Picklists Available for the Accreditation Request Object

Field Name
Status

Application

The application object is the means by which a new or existing partner applies for a partnership with the brand owner. The brand owner then reviews, approves, or rejects the application.

Child Objects

[Activity](#), [Book](#).

Methods Called

[Table 369](#) details the methods called by the Application service.

Table 369. Methods Called by Application Service

Method	Name as Defined in Service
"Delete" on page 105	ApplicationDelete
"Execute" on page 108	ApplicationExecute
"Insert" on page 112	ApplicationInsert
"QueryPage" on page 115	ApplicationQueryPage
"Update" on page 136	ApplicationUpdate

Fields

[Table 370](#) details the required and read-only fields for the application object.

Table 370. Required and Read-Only Fields for the Application Object

Field Name	Type
ContactFirstName	Required
ContactLastName	Required
SubmissionStatus	Required
CompanyName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

[Table 371](#) details the pick map fields for the application object.

Table 371. Pick Map Fields for the Application Object

Pick Map Field	Maps To
CurrentApproverAlias	CurrentApproverId
AssociatedPartnerName	PartnerId
PartnerProgramProgramName	PartnerProgramId
CampaignName	CampaignId
OwnerAlias	OwnerId
ProcessedByAlias	ProcessedById

[Table 372](#) provides a list of the user key combinations for the application object.

Table 372. User Key Fields on the Application Object

User Key Field Combinations
Id
ExternalSystemId
CompanyName

Table 373 details the picklists available for the application object.

Table 373. Picklists Available for the Application Object

Field Name
ApprovalStatus
PartnerType
ProgramLevel
RejectReason
Source
SubmissionStatus

Certification

The certification object stores information about an individual's competency in a specific set of skills. In Oracle CRM On Demand, certifications are always associated with an individual, in contrast to accreditations, which are always associated with a partner company. A certification is proof of an individual's competency in a specific set of skills.

Child Objects

[Accreditation](#), [Attachment](#), [Book](#), [Category](#), [Certification](#), [Certification Request](#), [Course](#), Custom Object 01-03, Custom Object 4 through 15, [Exam](#), [Product](#), [Solution](#).

Methods Called

Table 374 details the methods called by the Certification service.

Table 374. Methods Called by Certification Service

Method	Name as Defined in Service
"Delete" on page 105	CertificationDelete
"Execute" on page 108	CertificationExecute
"Insert" on page 112	CertificationInsert
"QueryPage" on page 115	CertificationQueryPage
"Update" on page 136	CertificationUpdate

Fields

[Table 375](#) details the required and read-only fields for the certification object.

Table 375. Required and Read-Only Fields for the Certification Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

[Table 376](#) details the pick map fields for the certification object.

Table 376. Pick Map Fields for the Certification Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
PrimaryProductCategoryName	PrimaryProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId

[Table 377](#) provides a list of the user key combinations for the certification object.

Table 377. User Key Fields on the Certification Object

User Key Field Combinations
Id
Name

[Table 378](#) details the picklists available for the certification object.

Table 378. Picklists Available for the Certification Object

Field Name
PaymentOption
Status
Type

Certification Request

The certification request object stores information about an individual's request for certification.

Child Objects

None.

Methods Called

[Table 379](#) details the methods called by the Certification Request service.

Table 379. Methods Called by Certification Request Service

Method	Name as Defined in Service
"Delete" on page 105	CertificationRequestDelete
"Execute" on page 108	CertificationRequestExecute
"Insert" on page 112	CertificationRequestInsert
"QueryPage" on page 115	CertificationRequestQueryPage
"Update" on page 136	CertificationRequestUpdate

Fields

[Table 380](#) details the required and read-only fields for the certification request object.

Table 380. Required and Read-Only Fields for the Certification Request Object

Field Name	Type
CertificationId	Required
ContactId	Required

Table 380. Required and Read-Only Fields for the Certification Request Object

Field Name	Type
OwnerId	Required
Audit Fields	Read-only

[Table 381](#) details the pick map fields for the certification request object.

Table 381. Pick Map Fields for the Certification Request Object

Pick Map Field	Maps To
CertificationName, CertificationNum, CertificationStatus, CertificationType, CertificationExternalSystemId	CertificationId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
PlanName, PlanType, PlanStatus, PlanDescription, PlanExternalSystemId	PlanId

[Table 382](#) provides a list of the user key combinations for the certification request object.

Table 382. User Key Fields on the Certification Request Object

User Key Field Combinations
Id
CertificationId, ContactId, and RequestDate

[Table 383](#) details the picklists available for the certification request object.

Table 383. Picklists Available for the Certification Request Object

Field Name
Status

Course

The course object stores information about a training course. A course is a structured educational programme offered with the primary aim of imparting a specific set of skills to an individual. Many companies, especially those that deal with complex products or services, require their partners and resellers to be adequately trained in and familiar with what they sell. Typically, courses focus on a specific product or product family; however, at times, they might also deal with generic concepts or principles. Courses might either be theoretical or practical, or a blend of both.

Child Objects

[Attachment](#), [Book](#), [Category](#), [Certification](#), [Course](#), [Course Enrollment](#), Custom Object 01-03, Custom Object 4 through 15, [Exam](#), [Partner](#), [Product](#), [Solution](#).

Methods Called

[Table 384](#) details the methods called by the Course service.

Table 384. Methods Called by Course Service

Method	Name as Defined in Service
"Delete" on page 105	CourseDelete
"Execute" on page 108	CourseExecute
"Insert" on page 112	CourseInsert
"QueryPage" on page 115	CourseQueryPage
"Update" on page 136	CourseUpdate

Fields

[Table 385](#) details the required and read-only fields for the course object.

Table 385. Required and Read-Only Fields for the Course Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

Table 386 details the pick map fields for the course object.

Table 386. Pick Map Fields for the Course Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
ProductCategoryName	ProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
ExamType, ExamStatus, ExamName, ExamId, ExamExternalSystemId	ExamId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 387 provides a list of the user key combinations for the course object.

Table 387. User Key Fields on the Course Object

User Key Field Combinations
Id
CourseId, CandidateId, and EnrollDate

Table 388 details the picklists available for the course object.

Table 388. Picklists Available for the Course Object

Field Name
Type
Status
Format
Medium
PaymentOption

Course Enrollment

The course enrollment object stores information about an individual's enrollment for a course.

Child Objects

None

Methods Called

Table 389 details the methods called by the Course Enrollment service.

Table 389. Methods Called by Course Enrollment Service

Method	Name as Defined in Service
"Delete" on page 105	CourseEnrollmentDelete
"Execute" on page 108	CourseEnrollmentExecute
"Insert" on page 112	CourseEnrollmentInsert
"QueryPage" on page 115	CourseEnrollmentQueryPage
"Update" on page 136	CourseEnrollmentUpdate

Fields

[Table 390](#) details the required and read-only fields for the course enrollment object.

Table 390. Required and Read-Only Fields for the Course Enrollment Object

Field Name	Type
CourseId	Required
CandidateId	Required
OwnerId	Required
Audit Fields	Read-only

[Table 391](#) details the pick map fields for the course enrollment object.

Table 391. Pick Map Fields for the Course Enrollment Object

Pick Map Field	Maps To
CourseName, CourseStatus, CourseType, CourseID, CourseExternalSystemId	CourseId
CandidateContactEmail, CandidateContactFirstName, CandidateContactLastName, CandidateContactFullName, CandidateIntegrationId, CandidateAccountName, CandidateExternalSystemId	CandidateId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
OfferingPartnerChannelAccountManagerAlias, OfferingPartnerIntegrationId, OfferingPartnerExternalSystemId, OfferingPartnerLocation, OfferingPartnerMainPhone, OfferingPartnerPartnerName	OfferingPartnerId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 392 provides a list of the user key combinations for the course enrollment object.

Table 392. User Key Fields on the Course Enrollment Object

User Key Field Combinations
CourseId
CandidateId
EnrollDate

Table 393 details the picklists available for the course enrollment object.

Table 393. Picklists Available for the Course Enrollment Object

Field Name
Status

Deal Registration

The deal registration object stores information about the process by which a partner company requests exclusive rights to an opportunity from the brand owner.

Child Objects

[Activity](#), [Book](#), Custom Object 01 -03, Custom Object 4 through 15, [Special Pricing Product](#).

Methods Called

Table 394 details the methods called by the Deal Registration service.

Table 394. Methods Called by Deal Registration Service

Method	Name as Defined in Service
"Delete" on page 105	DealRegistrationDelete
"Execute" on page 108	DealRegistrationExecute
"Insert" on page 112	DealRegistrationInsert
"QueryPage" on page 115	DealRegistrationQueryPage
"Update" on page 136	DealRegistrationUpdate

Fields

[Table 395](#) details the required and read-only fields for the deal registration object.

Table 395. Required and Read-Only Fields for the Deal Registration Object

Field Name	Type
DealRegistrationName	Required
Type	Required
PartnerId	Required
SubmissionStatus	Required
DealRegistrationId	Read-only
ConvertedtoOpportunity	Read-only
Audit Fields	Read-only

[Table 396](#) details the pick map fields for the deal registration object.

Table 396. Pick Map Fields for the Deal Registration Object

Pick Map Field	Maps To
PartnerExternalSystemId	PartnerId
PartnerName	PartnerId

[Table 397](#) provides a list of the user key combinations for the deal registration object.

Table 397. User Key Fields on the Deal Registration Object

User Key Field Combinations
ExternalSystemId
DealRegistrationName and PartnerId
Id

[Table 398](#) details the picklists available for the deal registration object.

Table 398. Picklists Available for the Deal Registration Object

Field Name
Type
Status

Table 398. Picklists Available for the Deal Registration Object

Field Name
State
Country
Salutation
ApprovalStatus
RejectReason

Exam

The exam object stores information about a training examination. Exams measure the competency of an individual in a specific set of skills. They are usually the next logical step for an individual after completing a course. Typically, a course culminates in an exam (or exams) and an individual is expected to register and appear for the exams upon completing the course. In fact, many companies consider that a course has been successfully completed only if the exams are passed with at least the minimum passing score.

Child Objects

[Attachment](#), [Book](#), [Category](#), [Certification](#), [Course](#), Custom Object 01-03, Custom Object 4 through 15, [Exam](#), [Exam Registration](#), [Partner](#), [Product](#), [Solution](#).

Methods Called

[Table 399](#) details the methods called by the Exam service.

Table 399. Methods Called by Exam Service

Method	Name as Defined in Service
"Delete" on page 105	ExamDelete
"Execute" on page 108	ExamExecute
"Insert" on page 112	ExamInsert
"QueryPage" on page 115	ExamQueryPage
"Update" on page 136	ExamUpdate

Fields

Table 400 details the required and read-only fields for the exam object.

Table 400. Required and Read-Only Fields for the Exam Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
Audit Fields	Read-only

Table 401 details the pick map fields for the exam object.

Table 401. Pick Map Fields for the Exam Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
ProductCategoryName	ProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

[Table 402](#) provides a list of the user key combinations for the exam object.

Table 402. User Key Fields on the Exam Object

User Key Field Combinations
Id
Name

[Table 403](#) details the picklists available for the exam object.

Table 403. Picklists Available for the Exam Object

Field Name
Type
Status
Format
Medium
PaymentOption
Grade

Exam Registration

The exam registration object stores information about an individual's registration for an examination.

Child Objects

None.

Methods Called

[Table 404](#) details the methods called by the Exam Registration service.

Table 404. Methods Called by Exam Registration Service

Method	Name as Defined in Service
"Delete" on page 105	ExamRegistrationDelete
"Execute" on page 108	ExamRegistrationExecute
"Insert" on page 112	ExamRegistrationInsert

Table 404. Methods Called by Exam Registration Service

Method	Name as Defined in Service
"QueryPage" on page 115	ExamRegistrationQueryPage
"Update" on page 136	ExamRegistrationUpdate

Fields

[Table 405](#) details the required and read-only fields for the exam registration object.

Table 405. Required and Read-Only Fields for the Exam Registration Object

Field Name	Type
ExamId	Required
CandidateId	Required
ExamDate	Required
OwnerId	Required
Audit Fields	Read-only

[Table 406](#) details the pick map fields for the exam registration object.

Table 406. Pick Map Fields for the Exam Registration Object

Pick Map Field	Maps To
ExamName, ExamStatus, ExamType, ExamID, ExamExternalSystemId	ExamId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
PartnerChannelAccountManagerAlias, PartnerIntegrationId, PartnerExternalSystemId, PartnerLocation, PartnerMainPhone, PartnerName,	PartnerId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr.	OwnerId

Table 407 provides a list of the user key combinations for the exam registration object.

Table 407. User Key Fields on the Exam Registration Object

User Key Field Combinations
Id
CourseId, CandidateId, OwnerId, and ExamDate

Table 408 details the picklists available for the exam registration object.

Table 408. Picklists Available for the Exam Registration Object

Field Name
Status

Fund

The fund object stores information about a sum of money or a set of other resources that are set aside for a specific purpose.

Child Objects

[Activity](#), Custom Object 01-03, Custom Object 4 through 15, [MDF Request](#), [Special Pricing Product](#).

Methods Called

Table 409 details the methods called by the Fund service.

Table 409. Methods Called by Fund Service

Method	Name as Defined in Service
"Delete" on page 105	FundDelete
"Execute" on page 108	FundExecute
"Insert" on page 112	FundInsert
"QueryPage" on page 115	FundQueryPage
"Update" on page 136	FundUpdate

Fields

[Table 410](#) details the required and read-only fields for the fund object.

Table 410. Required and Read-Only Fields for the Fund Object

Field Name	Type
FundName	Required
FundId	Required
TargetAmount	Required
Audit Fields	Read-only

[Table 411](#) details the pick map field for the fund object.

Table 411. Pick Map Field for the Fund Object

Pick Map Field	Maps To
Partner	PartnerId

[Table 412](#) provides a list of the user key combinations for the fund object.

Table 412. User Key Fields on the Fund Object

User Key Field Combinations
FundId
Id

[Table 413](#) details the picklists available for the fund object.

Table 413. Picklists Available for the Fund Object

Field Name
FundCode
Status
Type
CurrencyCode
Partner
Period

MDF Request

The MDF request object stores information about market development funds (MDF) requests.

Child Objects

[Activity](#), [Book](#), Custom Object 01-03, Custom Object 4 through 15.

Methods Called

[Table 414](#) details the methods called by the MDF Request service.

Table 414. Methods Called by MDF Request Service

Method	Name as Defined in Service
"Delete" on page 105	MDFRequestDelete
"Execute" on page 108	MDFRequestExecute
"Insert" on page 112	MDFRequestInsert
"QueryPage" on page 115	MDFRequestQueryPage
"Update" on page 136	MDFRequestUpdate

Fields

[Table 415](#) details the required and read-only fields for the MDF request object.

Table 415. Read-Only Fields on the MDF Request Object

Field Name	Type
RequestName	Required
PrincipalPartnerAccountId	Required
FundName	Required
DueDate	Required
SubmissionStatus	Required
MDFRequestId	Read-only
Audit Fields	Read-only

Table 416 details the pickmap fields for the MDF request object.

Table 416. Pick Map Fields for the MDF Request Object

Pick Map Field	Maps To
PrincipalPartnerAccountExternalSystemId	PrincipalPartnerAccountId
PrincipalPartnerAccountIntegrationId	PrincipalPartnerAccountId
PrincipalPartnerAccountPartnerName, PrincipalPartnerAccountLocation	PrincipalPartnerAccountId
PrincipalPartnerAccountChannelAccountManagerAlias	PrincipalPartnerAccountId

Table 417 provides a list of the user key combinations for the MDF request object.

Table 417. User Key Fields on the MDF Request Object

User Key Field Combinations
RequestName
Id
ExternalSystemId

Table 418 details the picklists available for the MDF request object.

Table 418. Picklists Available for the MDF Request Object

Field Name
Status
Type
SubmissionStatus
ApprovalStatus
MarketingPurpose
Region
Category
ReasonCode
ClaimStatus

Partner

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

Child Objects

[Application](#), [Book](#), [Business Plan](#), Custom Object 4 through 15, [Deal Registration](#), [MDF Request](#), [Partner](#), [Policy](#), [Shared Address](#), [Special Pricing Product](#).

Methods Called

[Table 419](#) details the methods called by the Partner service.

Table 419. Methods Called by Partner Service

Method	Name as Defined in Service
"Insert" on page 112	PartnerInsert
"QueryPage" on page 115	PartnerQueryPage
"Update" on page 136	PartnerUpdate

Fields

[Table 420](#) details the read-only fields for the partner object.

Table 420. Read-Only Fields on the Partner Object

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

[Table 421](#) details the pickmap fields for the partner object.

Table 421. Pick Map Fields for the Partner Object

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

Table 421. Pick Map Fields for the Partner Object

Pick Map Field	Maps To
SourceCampaignName	SourceCampaignId
OwnerPartnerAccountPartnerName	OwnerPartnerId
PrincipalPartnerPartnerName	PrincipalPartnerId
OriginatingPartnerAccountPartnerName	OriginatingPartnerId
Territory	TerritoryId

Table 422 provides a list of the user key combinations for the partner object.

Table 422. User Key Fields on the Partner Object

User Key Field Combinations
Id
ExternalSystemId
Location
PartnerName

Table 423 details the picklists available for the partner object.

Table 423. Picklists Available for the Partner Object

Field Name
CallFrequency
ComplianceStatus
Expertise
InfluenceType
MarketPotential
MarketSegment
PartnerOrgStatus
PartnerLevel
Priority
Region

Table 423. Picklists Available for the Partner Object

Field Name
Route
Status

Price List

The price list object store information about a price list for products.

Child Objects

[Account](#), [Partner](#), [Price List Line Item](#), [Special Pricing Product](#).

Methods Called

[Table 424](#) details the methods called by the Price List service.

Table 424. Methods Called by Price List Service

Method	Name as Defined in Service
"Delete" on page 105	PriceListDelete
"Execute" on page 108	PriceListExecute
"Insert" on page 112	PriceListInsert
"QueryPage" on page 115	PriceListQueryPage
"Update" on page 136	PriceListUpdate

Fields

[Table 425](#) details the required and read-only fields for the price list object.

Table 425. Required and Read-Only Fields for the Price List Object

Field Name	Type
PriceListName	Required
EffectiveFrom	Required
Type	Required
Status	Required

Table 425. Required and Read-Only Fields for the Price List Object

Field Name	Type
CurrencyCode	Required
Audit Fields	Read-only

There are no pick map fields for the price list object.

[Table 426](#) details the user keys for the price list object.

Table 426. User Keys for the Price List Object

Field Name
PriceListName and Type
Id
ExternalSystemId

[Table 427](#) details the picklists available for the price list object.

Table 427. Picklists Available for the Price List Object

Field Name
Type
Status

Price List Line Item

The price list line item object store information about a product in a price list.

Child Objects

None

Methods Called

Table 428 details the methods called by the Price List Line Item service.

Table 428. Methods Called by Price List Line Item Service

Method	Name as Defined in Service
“Delete” on page 105	PriceListLineItemDelete
“Execute” on page 108	PriceListLineItemExecute
“Insert” on page 112	PriceListLineItemInsert
“QueryPage” on page 115	PriceListLineItemQueryPage
“Update” on page 136	PriceListLineItemUpdate

Fields

Table 429 details the required and read-only fields for the price list line item object.

Table 429. Required and Read-Only Fields for the Price List Line Item Object

Field Name	Type
PriceListId	Required
ProductId	Required
Type	Required
Audit Fields	Read-only

Table 430 details the pick map field for the price list line item object.

Table 430. Pick Map Field for the Price List Line Item Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
ProductName, ProductCategory	ProductId
PriceListPriceListName	PriceListId
PriceListIntegrationId	PriceListId
PriceListExternalSystemId	PriceListId

Table 431 details the user keys for the price list line item object.

Table 431. User Keys for the Price List Line Item Object

Field Name
Id
ExternalSystemId
ProductId and PriceListId and PriceType

Table 432 details the picklists available for the price list line item object.

Table 432. Picklists Available for the Price List Line Item Object

Field Name
PriceType

Special Pricing Product

The special pricing product object stores information about a product associated with a special pricing request.

Child Objects

None.

Methods Called

Table 433 details the methods called by the Special Pricing Product service.

Table 433. Methods Called by Special Pricing Product Service

Method	Name as Defined in Service
"Delete" on page 105	SPRequestLineItemDelete
"Execute" on page 108	SPRequestLineItemExecute
"Insert" on page 112	SPRequestLineItemInsert
"QueryPage" on page 115	SPRequestLineItemQueryPage
"Update" on page 136	SPRequestLineItemUpdate

Fields

[Table 434](#) details the required and read-only fields for the special pricing product object.

Table 434. Read-Only Fields on the Special Pricing Product Object

Field Name	Type
ProductId	Required
SPRequestId	Read-only
Audit Fields	Read-only

[Table 435](#) details the pickmap fields for the special pricing product object.

Table 435. Pick Map Fields for the Special Pricing Product Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
ProductName, ProductCategory	ProductId

[Table 436](#) provides a list of the user key combinations for the special pricing product object.

Table 436. User Key Fields on the Special Pricing Product Object

User Key Field Combinations
Id
ExternalSystemId
ItemNumber

There are no picklists available for the special pricing product object.

Special Pricing Request

The Special Pricing (SP) request object stores information about a special pricing request, which is a set of business processes where a partner asks the brand owner for a discount on a product.

Child Objects

[Activity](#), [Book](#), Custom Object 01 - 03, Custom Object 4 through 15, [Special Pricing Product](#).

Methods Called

[Table 437](#) details the methods called by the SP Request service.

Table 437. Methods Called by SP Request Service

Method	Name as Defined in Service
"Delete" on page 105	SPRequestDelete
"Execute" on page 108	SPRequestExecute
"Insert" on page 112	SPRequestInsert
"QueryPage" on page 115	SPRequestQueryPage
"Update" on page 136	SPRequestUpdate

Fields

[Table 438](#) details the required and read-only fields for the SP request object.

Table 438. Read-Only Fields on the SP Request Object

Field Name	Type
SPRequestName	Required
PrincipalPartnerAccountId	Required
SubmissionStatus	Required
SPRequestId	Read-only
Audit Fields	Read-only

[Table 439](#) details the pickmap fields for the SP request object.

Table 439. Pick Map Fields for the SP Request Object

Pick Map Field	Maps To
PrincipalPartnerAccountExternalSystemId	PrincipalPartnerAccountId
PrincipalPartnerAccountIntegrationId	PrincipalPartnerAccountId

Table 439. Pick Map Fields for the SP Request Object

Pick Map Field	Maps To
PrincipalPartnerAccountPartnerName, PrincipalPartnerAccountLocation	PrincipalPartnerAccountId
PrincipalPartnerAccountChannelAccountManagerAlias	PrincipalPartnerAccountId

Table 440 provides a list of the user key combinations for the SP request object.

Table 440. User Key Fields on the SP Request Object

User Key Field Combinations
Id
ExternalSystemId
SPRequestName
FundId and RequestDate

Table 441 details the picklists available for the SP request object.

Table 441. Picklists Available for the SP Request Object

Field Name
SubmissionStatus
ApprovalStatus
RejectReason
Type
ClaimStatus

Life Sciences Edition Parent Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Life Sciences Edition. For these parent objects, all child objects are also accessible.

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

The following objects are detailed in this topic:

- ["Allocation" on page 510](#)
- ["Business Plan" on page 512](#)
- ["Blocked Product" on page 514](#)
- ["Contact Best Time" on page 515](#)
- ["Contact State License" on page 517](#)
- ["Inventory Audit Report" on page 519](#)
- ["Inventory Period" on page 521](#)
- ["MedEd" on page 522](#)
- ["Message Response" on page 524](#)
- ["Messaging Plan" on page 526](#)
- ["Messaging Plan Item" on page 528](#)
- ["Messaging Plan Item Relation" on page 530](#)
- ["Modification Tracking" on page 532](#)
- ["Objective" on page 533](#)
- ["Order" on page 536](#)
- ["Order Item" on page 537](#)
- ["Plan Account" on page 539](#)
- ["Plan Contact" on page 540](#)
- ["Plan Opportunity" on page 542](#)
- ["Sample Disclaimer" on page 544](#)
- ["Sample Inventory" on page 545](#)
- ["Sample Lot" on page 547](#)
- ["Sample Request" on page 549](#)
- ["Sample Request Item" on page 550](#)
- ["Sample Transaction" on page 552](#)

■ [“Signature” on page 554](#)

■ [“Transaction Item” on page 555](#)

NOTE: While the Business Plan, Objective, Plan Account, Plan Contact, and Plan Opportunity functionality was developed in the context of Oracle CRM On Demand Life Sciences Edition, all Oracle CRM On Demand customers can access and leverage this functionality. Your administrator or customer advocate can enable the Business Plan, Objective, Plan Account, Plan Contact, and Plan Opportunity objects and make these pages available for your use.

Allocation

The allocation object determines the period of time in which sample products are provided for a sales representative to distribute to clinics, hospitals, or physician contacts during a sample drop call execution.

Child Objects

None.

Methods Called

[Table 442](#) details the methods called by the Allocation service.

Table 442. Methods Called by Allocation Service

Method	Name as Defined in Service
“Delete” on page 105	AllocationDelete
“Execute” on page 108	AllocationExecute
“Insert” on page 112	AllocationInsert
“QueryPage” on page 115	AllocationQueryPage
“Update” on page 136	AllocationUpdate

Fields

[Table 443](#) details the required and read-only fields for the allocation object.

Table 443. Required and Read-Only Fields for the Allocation Object

Field Name	Type
AllocationType	Required
OwnerId	Required
ProductName	Required

Table 443. Required and Read-Only Fields for the Allocation Object

Field Name	Type
StopSampleFlag	Required
ActivePeriod	Read-only
Audit Fields	Read-only

[Table 444](#) details the pick map fields for the allocation object.

Table 444. Pick Map Fields for the Allocation Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 445](#) provides a list of the user key combinations for the allocation object.

Table 445. User Key Fields on the Allocation Object

User Key Field Combinations
Id
ExternalSystemId
OwnerId and ProductId and StartDate and EndDate and AllocationType

[Table 446](#) details the picklists available for the allocation object.

Table 446. Picklists Available for the Allocation Object

Field Name
AllocationType

Business Plan

The business plan object allows sales organizations to explicitly establish strategic goals (for example, revenue targets) and action plans to achieve those goals, in the form of objectives, activities, and opportunities.

Child Objects

[Activity](#), [Business Plan](#), [MedEd](#), [Messaging Plan](#), [Objective](#), [Plan Account](#), [Plan Contact](#), [Plan Opportunity](#).

Methods Called

[Table 447](#) details the methods called by the Business Plan service.

Table 447. Methods Called by Business Plan Service

Method	Name as Defined in Service
"Delete" on page 105	BusinessPlanDelete
"Execute" on page 108	BusinessPlanExecute
"Insert" on page 112	BusinessPlanInsert
"QueryPage" on page 115	BusinessPlanQueryPage
"Update" on page 136	BusinessPlanUpdate

Fields

[Table 448](#) details the required and read-only fields for the business plan object.

Table 448. Required and Read-Only Fields for the Business plan Object

Field Name	Type
PlanName	Required
PeriodName	Required
Type	Required
Status	Required
Audit Fields	Read-only

Table 449 details the pick map fields for the business plan object.

Table 449. Pick Map Fields for the Business plan Object

Pick Map Field	Maps To
ParentPlanNamePlanName	ParentPlanId
ParentPlanNameExternalSystemId	ParentPlanId
ParentPlanNameIntegrationId	ParentPlanId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
PeriodName	PeriodId
PeriodExternalSystemId	PeriodId
PeriodIntegrationId	PeriodId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 450 provides a list of the user key combinations for the business plan object.

Table 450. User Key Fields on the Business plan Object

User Key Field Combinations
Id
ExternalSystemId
PeriodId and PlanName

Table 451 details the picklists available for the business plan object.

Table 451. Picklists Available for the Business plan Object

Field Name
Type
Status

Blocked Product

The blocked product object specifies a product that is blocked for a particular contact. If a product is blocked for a contact, the contact is prevented from receiving any information, samples, or promotional items for the product.

Child Objects

None

Methods Called

[Table 452](#) details the methods called by the Blocked Product service.

Table 452. Methods Called by Blocked Product Service

Method	Name as Defined in Service
"Delete" on page 105	BlockedProductDelete
"Execute" on page 108	BlockedProductExecute
"Insert" on page 112	BlockedProductInsert
"QueryPage" on page 115	BlockedProductQueryPage
"Update" on page 136	BlockedProductUpdate

Fields

[Table 453](#) details the required and read-only fields for the blocked product object.

Table 453. Required and Read-Only Fields for the Blocked Product Object

Field Name	Type
ProductId	Required
ContactId	Required
Audit Fields	Read-only

Table 454 details the pick map fields for the blocked product object.

Table 454. Pick Map Fields for the Blocked Product Object

Pick Map Field	Maps To
Contact	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
Owner	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

Table 455 provides a list of the user key combinations for the blocked product object.

Table 455. User Key Fields on the Blocked Product Object

User Key Field Combinations
Id
ExternalSystemId

Contact Best Time

The contact best time object records and tracks the best time and day of week to call on a contact (by typically, a physician or another health care provider).

NOTE: It is recommended to access the contact best time object as a child of the contact object. Using the Contact Web service is preferred as best call time is within the context of a contact.

Child Objects

None.

Methods Called

Table 456 details the methods called by the Contact Best Time service.

Table 456. Methods Called by Contact Best Time Service

Method	Name as Defined in Service
"Delete" on page 105	ContactBestTimeDelete
"Execute" on page 108	ContactBestTimeExecute
"Insert" on page 112	ContactBestTimeInsert
"QueryPage" on page 115	ContactBestTimeQueryPage
"Update" on page 136	ContactBestTimePlanUpdate

Fields

Table 457 details the required and read-only fields for the contact best time object.

Table 457. Required and Read-Only Fields for the Contact Best Time Object

Field Name	Type
StartDay	Required
StartTime	Required
EndTime	Required
FullAddress	Required
ContactFullName	Required
Audit Fields	Read-only

Table 458 details the pick map fields for the contact best time object.

Table 458. Pick Map Fields for the Contact Best Time Object

Pick Map Field	Maps To
AddressExternalSystemId	AddressId
AddressIntegrationId	AddressId
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId

Table 459 provides a list of the user key combinations for the contact best time object.

Table 459. User Key Fields on the Contact Best Time Object

User Key Field Combinations
ContactId
AddressId
ExternalSystemId

Table 460 details the picklists available for the contact best time object.

Table 460. Picklists Available for the Contact Best Time Object

Field Name
StartDay

Contact State License

The contact state license object stores the medical license information for a contact, typically a physician.

Child Objects

None.

Methods Called

Table 461 details the methods called by the contact state license service.

Table 461. Methods Called by Contact State License Service

Method	Name as Defined in Service
"Delete" on page 105	ContactLicenseDelete
"Execute" on page 108	ContactLicenseExecute
"Insert" on page 112	ContactLicenseInsert
"QueryPage" on page 115	ContactLicenseQueryPage
"Update" on page 136	ContactLicenseUpdate

Fields

[Table 462](#) details the required and read-only fields for the contact state license object.

Table 462. Required and Read-Only Fields for the Contact State LicenseObject

Field Name	Type
ContactFullName	Required
LicenseNumber	Required
Audit Fields	Read-only

[Table 463](#) details the pick map fields for the contact state license object.

Table 463. Pick Map Fields for the Contact State License Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 464](#) provides a list of the user key combinations for the contact state license object.

Table 464. User Key Fields on the Contact State License Object

User Key Field Combinations
Id
ExternalSystemId
ContactId and State
LicenseNumber

[Table 465](#) details the picklists available for the contact state license object.

Table 465. Picklists Available for the Contact State License Object

Field Name
State
Status

Inventory Audit Report

The inventory audit report object stores information about the reports generated by Oracle CRM On Demand when a sales representative chooses to create a new interim or final audit count for an inventory period.

Child Objects

[Sample Inventory](#)

Methods Called

[Table 466](#) details the methods called by the Inventory Audit Report service.

Table 466. Methods Called by Inventory Audit Report Service

Method	Name as Defined in Service
"Delete" on page 105	InventoryAuditReportDelete
"Execute" on page 108	InventoryAuditReportExecute
"Insert" on page 112	InventoryAuditReportInsert
"QueryPage" on page 115	InventoryPeriodQueryPage
"Update" on page 136	InventoryPeriodUpdate

Fields

[Table 467](#) details the required and read-only fields for the inventory audit report object.

Table 467. Required and Read-Only Fields for the Inventory Audit Report Object

Field Name	Type
Type	Required
Status	Required

Table 467. Required and Read-Only Fields for the Inventory Audit Report Object

Field Name	Type
Reason	Required
CountCompletedOn	Required
InventoryPeriodStartDate	Required
OwnerUserSignId	Required
Audit Fields	Read-only

[Table 468](#) details the pick map fields for the inventory audit report object.

Table 468. Pick Map Field for the Inventory Audit Report Object

Pick Map Field	Maps To
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIdIntegrationId	InventoryPeriodId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 469](#) provides a list of the user key combinations for the inventory audit report object.

Table 469. User Key Fields on the Inventory Audit Report Object

User Key Field Combinations
Id
ExternalSystemId
InventoryPeriodId and OwnerId and CountCompletedOn
Type

Table 470 details the picklists available for the inventory audit report object.

Table 470. Picklists Available for the Inventory Audit Report Object

Field Name
Type
Status
Reason

Inventory Period

The inventory period object represents a time frame within which all movement of samples or promotional items (sample drops, receipts, disbursements, and inventory adjustments) can be tracked.

Child Objects

[Inventory Audit Report](#), [Sample Inventory](#), [Sample Transaction](#), [Sample Transaction](#).

Methods Called

Table 471 details the methods called by the Inventory Period service.

Table 471. Methods Called by Inventory Period Service

Method	Name as Defined in Service
"Delete" on page 105	InventoryPeriodDelete
"Execute" on page 108	InventoryPeriodExecute
"Insert" on page 112	InventoryPeriodInsert
"QueryPage" on page 115	InventoryPeriodQueryPage
"Update" on page 136	InventoryPeriodUpdate

Fields

[Table 472](#) details the required and read-only fields for the inventory period object.

Table 472. Required and Read-Only Fields for the Inventory Period Object

Field Name	Type
StartDate	Required
OwnerId	Required
ActiveFlg	Required
ReconciledFlg	Required
Audit Fields	Read-only

[Table 473](#) details the pick map fields for the inventory period object.

Table 473. Pick Map Field for the Inventory Period Object

Pick Map Field	Maps To
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 474](#) provides a list of the user key combinations for the inventory period object.

Table 474. User Key Fields on the Inventory Period Object

User Key Field Combinations
Id
ExternalSystemId
OwnerId and StartDate

MedEd

The MedEd object allows you to plan and track medical education events. A medical education event can be as simple as a lunch-and-learn session in a physician's office or as complex as a seminar series or national sales meeting.

Child Objects

[Book](#), [Fund](#), [Invitee](#)

Methods Called

[Table 475](#) details the methods called by the MedEd service.

Table 475. Methods Called by MedEd Service

Method	Name as Defined in Service
"Delete" on page 105	MedEdDelete
"Execute" on page 108	MedEdExecute
"Insert" on page 112	MedEdInsert
"QueryPage" on page 115	MedEdQueryPage
"Update" on page 136	MedEdUpdate

Fields

[Table 476](#) details the required and read-only fields for the MedEd object.

Table 476. Read-Only Fields on the MedEd Object

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

[Table 477](#) details the pickmap fields for the MedEd object.

Table 477. Pick Map Fields for the MedEd Object

Pick Map Field	Maps To
ProductExternalId	ProductId
ProductIntegrationId	ProductId

Table 478 provides a list of the user key combinations for the MedEd object.

Table 478. User Key Fields on the MedEd Object

User Key Field Combinations
MedEdId
ExternalSystemId
Name

Table 479 details the picklists available for the MedEd object.

Table 479. Picklists Available for the MedEd Object

Field Name
EventStatusCode
EventTypeCode

Message Response

The message response object stores feedback received from the audience during the presentation of a message plan during a particular period.

Child Objects

None

Methods Called

Table 480 details the methods called by the Message Response service.

Table 480. Methods Called by Message Response Service

Method	Name as Defined in Service
"Delete" on page 105	MessageResponseDelete
"Execute" on page 108	MessageResponseExecute
"Insert" on page 112	MessageResponseInsert
"QueryPage" on page 115	MessageResponseQueryPage
"Update" on page 136	MessageResponseUpdate

Fields

[Table 481](#) details the required and read-only fields for the message response object.

Table 481. Read-Only Fields on the Message Response Object

Field Name	Type
EndTime	Required
FollowUp	Required
Response	Required
SequenceNumber	Required
SolutionId	Required
StartTime	Required
Message	Read-only
Duration	Read-only
Audit Fields	Read-only

[Table 482](#) details the pickmap fields for the message response object.

Table 482. Pick Map Fields for the Message Response Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
AccountName	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
SolutionTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
PlanName	PlanId
PlanExternalSystemId	PlanId
PlanIntegrationId	PlanId
PlanItemName	PlanItemId

Table 482. Pick Map Fields for the Message Response Object

Pick Map Field	Maps To
PlanItemExternalSystemId	PlanItemId
PlanItemIntegrationId	PlanItemId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 483](#) provides a list of the user key combinations for the message response object.

Table 483. User Key Fields on the Message Response Object

User Key Field Combinations
Id
ExternalSystemId
PlanName and SolutionName

[Table 484](#) details the picklists available for the message response object.

Table 484. Picklists Available for the Message Response Object

Field Name
Response

Messaging Plan

The messaging plan object stores information about a presentation delivered using a personal computer or tablet computer for sales users to detail products and track customer feedback.

Child Objects

[Book](#), [Message Response](#), [Messaging Plan Item](#).

Methods Called

[Table 485](#) details the methods called by the Messaging Plan service.

Table 485. Methods Called by Messaging Plan Service

Method	Name as Defined in Service
"Delete" on page 105	MessagePlanDelete
"Execute" on page 108	MessagePlanExecute
"Insert" on page 112	MessagePlanInsert
"QueryPage" on page 115	MessagePlanQueryPage
"Update" on page 136	MessagePlanUpdate

Fields

[Table 486](#) details the required and read-only fields for the messaging plan object.

Table 486. Read-Only Fields on the Messaging Plan Object

Field Name	Type
Name	Required
OwnerId	Required
Type	Required
ProductName	Required
Status	Required
LockSequence	Required
DisclosureMandatory	Required
EnableFollowUp	Required
VerificationStatus	Read-only
Audit Fields	Read-only

[Table 487](#) details the pickmap fields for the messaging plan object.

Table 487. Pick Map Fields for the Messaging Plan Object

Pick Map Field	Maps To
SolutionTitle	SolutionId
SolutionExternalSystemId	SolutionId

Table 487. Pick Map Fields for the Messaging Plan Object

Pick Map Field	Maps To
SolutionIntegrationId	SolutionId
EffectivePeriodName	EffectivePeriodId
EffectivePeriodExternalSystemId	EffectivePeriodId
EffectivePeriodIntegrationId	EffectivePeriodId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 488](#) provides a list of the user key combinations for the messaging plan object.

Table 488. User Key Fields on the Messaging Plan Object

User Key Field Combinations
Id
ExternalSystemId
Name and OwnerId

[Table 489](#) details the picklists available for the messaging plan object.

Table 489. Picklists Available for the Messaging Plan Object

Field Name
Type
Status
Skin

Messaging Plan Item

The messaging plan item object stores information about the items that make up a message plan.

Child Objects

[Messaging Plan Item Relation](#), [Message Response](#).

Methods Called

[Table 490](#) details the methods called by the Messaging Plan Item service.

Table 490. Methods Called by Messaging Plan Item Service

Method	Name as Defined in Service
"Delete" on page 105	MsgPlanItemDelete
"Execute" on page 108	MsgPlanItemExecute
"Insert" on page 112	MsgPlanItemInsert
"QueryPage" on page 115	MsgPlanItemQueryPage
"Update" on page 136	MsgPlanItemUpdate

Fields

[Table 491](#) details the required and read-only fields for the messaging plan item object.

Table 491. Read-Only Fields on the Messaging Plan Item Object

Field Name	Type
SequenceNumber	Required
DisclosureMessage	Required
Type	Required
SolutionNameTitle	Required
ParentMessagePlanName	Required
Audit Fields	Read-only

[Table 492](#) details the pickmap fields for the messaging plan item object.

Table 492. Pick Map Fields for the Messaging Plan Item Object

Pick Map Field	Maps To
ParentMessagePlanName	ParentMessagePlanId
ParentMessagePlanExternalSystemId	ParentMessagePlanId
ParentMessagePlanIntegrationId	ParentMessagePlanId

Table 492. Pick Map Fields for the Messaging Plan Item Object

Pick Map Field	Maps To
SolutionNameTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 493](#) provides a list of the user key combinations for the messaging plan item object.

Table 493. User Key Fields on the Messaging Plan Item Object

User Key Field Combinations
Id
ExternalSystemId
ParentMessagePlanId and SolutionNameId and SequenceNumber and Type

[Table 494](#) details the picklists available for the messaging plan item object.

Table 494. Picklists Available for the Messaging Plan Item Object

Field Name
Type
Status
Rating

Messaging Plan Item Relation

The messaging plan item relation object stores information about a presentation item that provides optional information to support the primary message plan item.

Child Objects

None

Methods Called

Table 495 details the methods called by the Messaging Plan Item Relation service.

Table 495. Methods Called by Messaging Plan Item Service

Method	Name as Defined in Service
“Delete” on page 105	MsgPlanItemRelationDelete
“Execute” on page 108	MsgPlanItemRelationExecute
“Insert” on page 112	MsgPlanItemRelationInsert
“QueryPage” on page 115	MsgPlanItemRelationQueryPage
“Update” on page 136	MsgPlanItemRelationUpdate

Fields

Table 496 details the required and read-only fields for the messaging plan item relation object.

Table 496. Read-Only Fields on the Messaging Plan Item Relation Object

Field Name	Type
Type	Required
ParentMPItemName	Required
SolutionNameTitle	Required
Audit Fields	Read-only

Table 497 details the pickmap fields for the messaging plan item relation object.

Table 497. Pick Map Fields for the Messaging Plan Item Relation Object

Pick Map Field	Maps To
ParentMPItemName	ParentMPItemId
ParentMPItemExternalSystemId	ParentMPItemId
ParentMPItemIntegrationId	ParentMPItemid
SolutionNameTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
OwnerUserSignId	OwnerId

Table 497. Pick Map Fields for the Messaging Plan Item Relation Object

Pick Map Field	Maps To
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 498](#) provides a list of the user key combinations for the messaging plan item relation object.

Table 498. User Key Fields on the Messaging Plan Item Relation Object

User Key Field Combinations
Id
ExternalSystemId
ParentMPId and SolutionNameId and Type

[Table 499](#) details the picklists available for the messaging plan item relation object.

Table 499. Picklists Available for the Messaging Plan Item Relation Object

Field Name
Type
Status
Rating

Modification Tracking

The modification tracking object stores information about change events for a given set of Oracle CRM On Demand objects. When the modification tracking feature is enabled, for each event a modification record is written to the modification tracking log, which is used for synchronization of external mobile applications.

NOTE: The Modification Tracking Web service is for use with the Oracle CRM On Demand Disconnected Mobile Sales for Tablets.

Child Objects

None.

Methods Called

Table 500 details the methods called by the Modification Tracking service.

Table 500. Methods Called by Modification Tracking Service

Method	Name as Defined in Service
“Delete” on page 105	ModificationTrackingDelete
“Execute” on page 108	ModificationTrackingExecute
“Insert” on page 112	ModificationTrackingInsert
“QueryPage” on page 115	ModificationTrackingQueryPage
“Update” on page 136	ModificationTrackingUpdate

Fields

Table 501 details the required and read-only fields for the modification tracking object.

Table 501. Required and Read-Only Fields for the Modification Tracking Object

Field Name	Type
ObjectId	Required
EventName	Required
ObjectName	Required
ModificationNumber	Required
Audit Fields	Read-only

Table 502 provides a list of the user key combinations for the modification tracking object.

Table 502. User Key Fields on the Modification Tracking Object

User Key Field Combinations
Id
ObjectName

Objective

The objective object stores information about a tactical goal you want to achieve to attain a higher level strategic goal in a business plan.

Child Objects

[Activity](#), [Objective](#), [Opportunity](#).

Methods Called

[Table 503](#) details the methods called by the objective service.

Table 503. Methods Called by Objective Service

Method	Name as Defined in Service
"Delete" on page 105	ObjectiveDelete
"Execute" on page 108	ObjectiveExecute
"Insert" on page 112	ObjectiveInsert
"QueryPage" on page 115	ObjectiveQueryPage
"Update" on page 136	ObjectiveUpdate

Fields

[Table 504](#) details the required and read-only fields for the objective object.

Table 504. Required and Read-Only Fields for the Objective Object

Field Name	Type
ObjectiveName	Required
Type	Required
Status	Required
PeriodName	Required
Audit Fields	Read-only

[Table 505](#) details the pick map fields for the objective object.

Table 505. Pick Map Fields for the Objective Object

Pick Map Field	Maps To
ParentObjectiveObjectiveName	ParentObjectiveId
ParentObjectiveExternalSystemId	ParentObjectiveId
ParentObjectiveIntegrationId	ParentObjectiveId
PlanNamePlanName	PlanNameId

Table 505. Pick Map Fields for the Objective Object

Pick Map Field	Maps To
PlanNameExternalSystemId	PlanNameId
PlanNameIntegrationId	PlanNameId
AccountNameAccountName	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
ContactNameContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
ProductNameName	ProductId
ProductdExternalSystemId	ProductId
ProductIntegrationId	ProductId
PeriodName	PeriodId
PeriodExternalSystemId	PeriodId
PeriodIntegrationId	PeriodId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 506 provides a list of the user key combinations for the objective object.

Table 506. User Key Fields on the Objective Object

User Key Field Combinations
Id
ExternalSystemId
AccountNameId and ContactNameId and ProductNameId and PeriodId
ObjectiveName

[Table 507](#) details the picklists available for the objective object.

Table 507. Picklists Available for the Objective Object

Field Name
Status
Type
ObjectiveUnits

Order

The order object is used to track orders and associated order items for products. For example, sales representatives can create a direct request for sample fulfillment as well as submit an order to the main office for samples to be sent directly to customers.

Usage

For access to the order object and Order service, the Order Usage value on the company profile page must be set to CRM On Demand Order Management, and your user role must include the Enable Order Access privilege.

Child Objects

[Order Item](#)

Methods Called

[Table 508](#) details the methods called by the Order service.

Table 508. Methods Called by Order Service

Method	Name as Defined in Service
"Delete" on page 105	OrderDelete
"Execute" on page 108	OrderExecute
"Insert" on page 112	OrderInsert
"QueryPage" on page 115	OrderQueryPage
"Update" on page 136	OrderUpdate

Fields

[Table 509](#) details the required and read-only fields for the order object.

Table 509. Required and Read-Only Fields for the Order Object

Field Name	Type
OrderNumber	Read-only
BusinessAccountAccountName	Read-only
FinalAccountAccountName	Read-only
OpportunityName	Read-only
OrderIntegrationMessage	Read-only
OrderCreationStatus	Read-only
OrderCreationDate	Read-only
OwnerId	Read-only
Audit Fields	Read-only

[Table 510](#) details the pick map fields for the order object.

Table 510. Pick Map Fields for the Order Object

Pick Map Field	Maps To
ContactFullName	ContactId

[Table 511](#) provides a list of the user key combinations for the order object.

Table 511. User Key Fields on the Order Object

User Key Field Combinations
Id
OrderNumber

There are no picklists available for the Order object.

Order Item

The order item object tracks which products have been purchased by the customer as part of an order.

Usage

For access to the order item object and Order Item service, the Order Usage value on the company profile page must be set to CRM On Demand Order Management, and your user role must include the Enable Order Access privilege.

Methods Called

[Table 512](#) details the methods called by the Order Item service.

Table 512. Methods Called by Order Item Service

Method	Name as Defined in Service
"Delete" on page 105	OrderItemDelete
"Execute" on page 108	OrderItemExecute
"Insert" on page 112	OrderItemInsert
"QueryPage" on page 115	OrderItemQueryPage
"Update" on page 136	OrderItemUpdate

Fields

[Table 513](#) details the required and read-only fields for the order item object.

Table 513. Required and Read-Only Fields for the Order Item Object

Field Name	Type
OrderItemNum	Required
Quantity	Required
ProductName	Required
OrderNumber	Required
Audit Fields	Read-only

[Table 514](#) details the pick map fields for the order item object.

Table 514. Pick Map Fields for the Order Item Object

Pick Map Field	Maps To
ProductName	ProductId
OrderNumber	OrderId

Table 515 provides a list of the user key combinations for the order item object.

Table 515. User Key Fields on the Order Item Object

User Key Field Combinations
OrderItemNum

There are no picklists available for the order item object.

Plan Account

The plan account object is a mapping of accounts and business plans.

Child Objects

None.

Methods Called

Table 516 details the methods called by the plan account service.

Table 516. Methods Called by Plan Account Service

Method	Name as Defined in Service
"Delete" on page 105	AccountBusinessPlanDelete
"Execute" on page 108	AccountBusinessPlanExecute
"Insert" on page 112	AccountBusinessPlanInsert
"QueryPage" on page 115	AccountBusinessPlanQueryPage
"Update" on page 136	AccountBusinessPlanUpdate

Fields

Table 517 details the required and read-only fields for the plan account object.

Table 517. Required and Read-Only Fields for the Plan Account Object

Field Name	Type
AccountIDAccountName	Required
BusinessPlanPlanName	Required
ContactId	Required

Table 517. Required and Read-Only Fields for the Plan Account Object

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

[Table 518](#) details the pick map fields for the plan account object.

Table 518. Pick Map Fields for the Plan Account Object

Pick Map Field	Maps To
AccountIDAccountName	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
BusinessPlanPlanName	BusinessPlanId
BusinessPlanExternalSystemId	BusinessPlanId
BusinessPlanIntegrationId	BusinessPlanId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 519](#) provides a list of the user key combinations for the plan account object.

Table 519. User Key Fields on the Plan Account Object

User Key Field Combinations
Id
ExternalSystemId
BusinessPlanId and AccountIDId
BusinessPlanPlanName

Plan Contact

The plan contact object is a mapping of contacts and business plans.

Child Objects

None

Methods Called

[Table 520](#) details the methods called by the plan contact service.

Table 520. Methods Called by Plan Contact Service

Method	Name as Defined in Service
"Delete" on page 105	ContactBusinessPlanDelete
"Execute" on page 108	ContactBusinessPlanExecute
"Insert" on page 112	ContactBusinessPlanInsert
"QueryPage" on page 115	ContactBusinessPlanQueryPage
"Update" on page 136	ContactBusinessPlanUpdate

Fields

[Table 521](#) details the required and read-only fields for the plan contact object.

Table 521. Required and Read-Only Fields for the Plan Contact Object

Field Name	Type
ContactId	Required
BusinessPlanPlanName	Required
Audit Fields	Read-only

[Table 522](#) details the pick map fields for the plan contact object.

Table 522. Pick Map Fields for the Plan Contact Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
BusinessPlanPlanName	BusinessPlanId
BusinessPlanExternalSystemId	BusinessPlanId
BusinessPlanIntegrationId	BusinessPlanId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 523 provides a list of the user key combinations for the plan contact object.

Table 523. User Key Fields on the Plan Contact Object

User Key Field Combinations
Id
ExternalSystemId
BusinessPlanId and ContactId
ContactFullName

Plan Opportunity

The plan opportunity object is a mapping of opportunities and business plans.

Child Objects

None.

Methods Called

Table 524 details the methods called by the Plan Opportunity service.

Table 524. Methods Called by Plan Opportunity Service

Method	Name as Defined in Service
"Delete" on page 105	PlanOpportunityDelete
"Execute" on page 108	PlanOpportunityExecute
"Insert" on page 112	PlanOpportunityInsert
"QueryPage" on page 115	PlanOpportunityQueryPage
"Update" on page 136	PlanOpportunityUpdate

Fields

[Table 525](#) details the required and read-only fields for the plan opportunity object.

Table 525. Required and Read-Only Fields for the Plan Opportunity Object

Field Name	Type
OpportunityName	Required
PlanName	Required
Audit Fields	Read-only

[Table 526](#) details the pick map fields for the plan opportunity object.

Table 526. Pick Map Fields for the Plan Opportunity Object

Pick Map Field	Maps To
OpportunityName	OpportunityId
OpportunityExternalSystemId	OpportunityId
OpportunityIntegrationId	OpportunityId
PlanName	PlanId
PlanExternalSystemId	PlanId
PlanIntegrationId	PlanId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 527](#) provides a list of the user key combinations for the plan opportunity object.

Table 527. User Key Fields on the Plan Opportunity Object

User Key Field Combinations
Id
ExternalSystemId
PlanId and OpportunityId
OpportunityName

Sample Disclaimer

The sample disclaimer object stores the text that is displayed to deny responsibility for certain adverse effects and reduce liability for a product sample that is dropped off to a physician or other health care professional.

Child Objects

None

Methods Called

[Table 528](#) details the methods called by the Sample Disclaimer service.

Table 528. Methods Called by Sample Disclaimer Service

Method	Name as Defined in Service
"Delete" on page 105	SignatureDisclaimerDelete
"Execute" on page 108	SignatureDisclaimerExecute
"Insert" on page 112	SignatureDisclaimerInsert
"QueryPage" on page 115	SignatureDisclaimerQueryPage
"Update" on page 136	SignatureDisclaimerUpdate

Fields

[Table 529](#) details the required and read-only fields for the sample disclaimer object.

Table 529. Required and Read-Only Fields for the Sample Disclaimer Object

Field Name	Type
DisclaimerText	Required
Status	Required
Audit Fields	Read-only

Table 530 details the pick map field for the sample disclaimer object.

Table 530. Pick Map Field for the Sample Disclaimer Object

Pick Map Field	Maps To
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 531 details the user keys for the sample disclaimer object.

Table 531. User Keys for the Sample Disclaimer Object

Field Name
Id
ExternalSystemId
DisclaimerNumber
Status and Created

Table 532 details the picklists available for the sample disclaimer object.

Table 532. Picklists Available for the Sample Disclaimer Object

Field Name
Status

Sample Inventory

The sample inventory object stores information about a product that has been categorized as qualifying for dropping a sample.

Child Objects

None

Methods Called

[Table 533](#) details the methods called by the Sample Inventory service.

Table 533. Methods Called by Sample Inventory Service

Method	Name as Defined in Service
“Delete” on page 105	SampleInventoryDelete
“Execute” on page 108	SampleInventoryExecute
“Insert” on page 112	SampleInventoryInsert
“QueryPage” on page 115	SampleInventoryQueryPage
“Update” on page 136	SampleInventoryUpdate

Fields

[Table 534](#) details the required and read-only fields for the sample inventory object.

Table 534. Required and Read-Only Fields for the Sample Inventory Object

Field Name	Type
InventoryPeriodStartDate	Required
ProductName	Required
OpeningBalance	Required
SystemCount	Read-only
Difference	Read-only
InventoryLot	Read-only
Audit Fields	Read-only

[Table 535](#) details the pick map field for the sample inventory object.

Table 535. Pick Map Fields for the Sample Inventory Object

Pick Map Field	Maps To
LotNumLotId	LotNumId
LotNumExternalSystemId	LotNumId
LotNumIntegrationId	LotNumId
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIntegrationId	InventoryPeriodId

Table 535. Pick Map Fields for the Sample Inventory Object

Pick Map Field	Maps To
AuditReportExternalSystemId	AuditReportId
AuditReportIntegrationId	AuditReportId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 536](#) details the user keys for the sample inventory object.

Table 536. User Keys for the Sample Inventory Object

Field Name
SampleInventoryId
IntegrationId
ExternalSystemId
OwnerId and ProductId and InventoryPeriodId and LotNumId and AuditReportId
ProductName

Sample Lot

The sample lot object records information about lot numbers for sample products.

Child Objects

[Sample Inventory](#), [Transaction Item](#).

Methods Called

Table 537 details the methods called by the Sample Lot service.

Table 537. Methods Called by Sample Lot Service

Method	Name as Defined in Service
“Delete” on page 105	SampleLotDelete
“Execute” on page 108	SampleLotExecute
“Insert” on page 112	SampleLotInsert
“QueryPage” on page 115	SampleLotQueryPage
“Update” on page 136	SampleLotUpdate

Fields

Table 538 details the required and read-only fields for the sample lot object.

Table 538. Required and Read-Only Fields for the Sample Lot Object

Field Name	Type
LotId	Required
SampleName	Required
ExpirationDate	Required
Orderable	Required
ParentSampleNameName	Required
Status	Read-only
CutoffDate	Read-only
Audit Fields	Read-only

Table 539 details the pick map field for the sample lot object.

Table 539. Pick Map Field for the Sample Lot Object

Pick Map Field	Maps To
ParentSampleNameName	ParentSampleNameId
ParentSampleNameExternalSystemId	ParentSampleNameId
ParentSampleNameIntegrationId	ParentSampleNameId
OwnerUserId	OwnerId

Table 539. Pick Map Field for the Sample Lot Object

Pick Map Field	Maps To
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 540](#) details the user keys for the sample lot object.

Table 540. User Keys for the Sample Lot Object

Field Name
Id
ExternalSystemId
LotId

Sample Request

The sample request object is used to track sample requests and associated sample request items for products.

Child Objects

[Sample Request Item](#)

Methods Called

[Table 541](#) details the methods called by the Sample Request service.

Table 541. Methods Called by Sample Request Service

Method	Name as Defined in Service
"Delete" on page 105	SampleRequestDelete
"Execute" on page 108	SampleRequestExecute
"Insert" on page 112	SampleRequestInsert
"QueryPage" on page 115	SampleRequestQueryPage
"Update" on page 136	SampleRequestUpdate

Fields

[Table 542](#) details the required and read-only fields for the sample request object.

Table 542. Required and Read-Only Fields for the Sample Request Object

Field Name	Type
OrderNumber	Read-only
OrderIntegrationMessage	Read-only
OrderCreationDate	Read-only
OwnerId	Read-only
Audit Fields	Read-only

[Table 543](#) details the pick map fields for the sample request object.

Table 543. Pick Map Fields for the Sample Request Object

Pick Map Field	Maps To
OrderNumber	OrderId

[Table 544](#) provides a list of the user key combinations for the sample request object.

Table 544. User Key Fields on the Sample Request Object

User Key Field Combinations
Id
OrderNumber

The OrderCreationStatus picklist is available for the Sample Request object.

Sample Request Item

The sample request item object tracks which products have been purchased by the customer as part of a sample request.

Methods Called

[Table 545](#) details the methods called by the Sample Request Item service.

Table 545. Methods Called by Sample Request Item Service

Method	Name as Defined in Service
"Delete" on page 105	SampleRequestItemDelete
"Execute" on page 108	SampleRequestItemExecute
"Insert" on page 112	SampleRequestItemInsert
"QueryPage" on page 115	SampleRequestItemQueryPage
"Update" on page 136	SampleRequestItemUpdate

Fields

[Table 546](#) details the required and read-only fields for the sample request item object.

Table 546. Required and Read-Only Fields for the Sample Request Item Object

Field Name	Type
Quantity	Required
ProductName	Required
SampleRequestOrderNumber	Required
Audit Fields	Read-only

[Table 547](#) details the pick map fields for the sample request item object.

Table 547. Pick Map Fields for the Sample Request Item Object

Pick Map Field	Maps To
ProductName	ProductId

[Table 548](#) provides a list of the user key combinations for the sample request item object.

Table 548. User Key Fields on the Sample Request Item Object

User Key Field Combinations
OrderItemNum

There are no picklists available for the sample request item object.

Sample Transaction

The sample transaction object is a record for a sample transfer, receipt, disbursement, or inventory adjustment. A sample transaction is created so that all movement of samples and promotional items can be reported and reconciled at the end of an inventory period, and a running count of inventory products can be maintained during the life of an inventory period.

Child Objects

[Transaction Item](#)

Methods Called

[Table 549](#) details the methods called by the Sample Transaction service.

Table 549. Methods Called by Sample Transaction Service

Method	Name as Defined in Service
"Delete" on page 105	TransactionDelete
"Execute" on page 108	TransactionExecute
"Insert" on page 112	TransactionInsert
"QueryPage" on page 115	TransactionQueryPage
"Update" on page 136	TransactionUpdate

Fields

[Table 550](#) details the required and read-only fields for the sample transaction object.

Table 550. Required and Read-Only Fields for the Sample Transaction Object

Field Name	Type
Name	Required
TransactionDate	Required
TransactionType	Required
InvoiceNumber	Required
Audit Fields	Read-only

Table 551 details the pick map field for the sample transaction object.

Table 551. Pick Map Field for the Sample Transaction Object

Pick Map Field	Maps To
SalesPersonIDId	SalesPersonIDId
SalesPersonIDExternalSystemId	SalesPersonIDId
SalesPersonIDIntegrationId	SalesPersonIDId
TransferToUserSignId	TransferToId
TransferToExternalSystemId	TransferToId
TransferToIntegrationId	TransferToId
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIntegrationId	InventoryPeriodId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 552 details the user keys for the sample transaction object.

Table 552. User Keys for the Sample Transaction Object

Field Name
Id
ExternalSystemId
Name

Table 553 details the picklists available for the sample transaction object.

Table 553. Picklists Available for the Sample Transaction Object

Field Name
TransactionType
TransactionStatus
Reason

Signature

The signature object stores information about a signature associated with a call activity for samples that are dropped.

Child Objects

None

Methods Called

[Table 554](#) details the methods called by the Signature service.

Table 554. Methods Called by Signature Service

Method	Name as Defined in Service
"Delete" on page 105	SignatureDelete
"Execute" on page 108	SignatureExecute
"Insert" on page 112	SignatureItemInsert
"QueryPage" on page 115	SignatureQueryPage
"Update" on page 136	SignatureUpdate

Fields

[Table 555](#) details the required and read-only fields for the signature object.

Table 555. Required and Read-Only Fields for the Signature Object

Field Name	Type
ActivityId	Required
SignatureCtrl	Required
Audit Fields	Read-only

[Table 556](#) details the pick map field for the signature object.

Table 556. Pick Map Field for the Signature Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId

Table 556. Pick Map Field for the Signature Object

Pick Map Field	Maps To
ContactIntegrationId	ContactId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 557](#) details the user keys for the signature object.

Table 557. User Keys for the Signature Object

Field Name
Id
ExternalSystemId
ActivityId

Transaction Item

The transaction item object stores the details of the sample or promotional items associated with a sample transaction.

Child Objects

None

Methods Called

[Table 558](#) details the methods called by the Transaction Item service.

Table 558. Methods Called by Transaction Item Service

Method	Name as Defined in Service
"Delete" on page 105	TransactionItemDelete
"Execute" on page 108	TransactionItemExecute
"Insert" on page 112	TransactionItemInsert
"QueryPage" on page 115	TransactionItemQueryPage
"Update" on page 136	TransactionItemUpdate

Fields

[Table 559](#) details the required and read-only fields for the transaction item object.

Table 559. Required and Read-Only Fields for the Transaction Item Object

Field Name	Type
LineNumber	Required
SampleTransactionName	Required
ProductName	Required
Quantity	Required
Audit Fields	Read-only

[Table 560](#) details the pick map field for the transaction item object.

Table 560. Pick Map Field for the Transaction Item Object

Pick Map Field	Maps To
SampleTransactionName	SampleTransactionId
SampleTransactionExternalSystemId	SampleTransactionId
SampleTransactionIntegrationId	SampleTransactionId
LotNumLotId	LotNumId
LotNumExternalSystemId	LotNumId
LotNumIntegrationId	LotNumId
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIntegrationId	InventoryPeriodId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 561 details the user keys for the transaction item object.

Table 561. User Keys for the Transaction Item Object

Field Name
Id
ExternalSystemId
SampleTransactionId and LineNumber
ProductName

Financial Services Edition Parent Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Financial Services Edition. For these parent objects, all child objects are also accessible.

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

The following objects are detailed in this topic:

- "Claim" on page 558
- "Coverage" on page 560
- "Damage" on page 562
- "Financial Account" on page 563
- "Financial Account Holder" on page 565
- "Financial Account Holding" on page 566
- "Financial Plan" on page 568
- "Financial Product" on page 570
- "Financial Transaction" on page 571
- "Household" on page 573
- "Insurance Property" on page 575
- "Involved Party" on page 577
- "Policy" on page 578
- "Policy Holder" on page 580
- "Portfolio" on page 582

Claim

The claim object allows you to define and record details about an insurance policy claim. Claims are typically the claims of a contact or business that an insurance company manages.

Child Objects

[Activity](#), [Damage](#), [Involved Party](#), [Service Request](#).

Methods Called

[Table 562](#) details the methods called by the Claim service.

Table 562. Methods Called by Claim Service

Method	Name as Defined in Service
"Delete" on page 105	ClaimDelete
"Execute" on page 108	ClaimExecute
"Insert" on page 112	ClaimInsert
"QueryPage" on page 115	ClaimQueryPage
"Update" on page 136	ClaimUpdate

Fields

[Table 563](#) details the required and read-only fields for the claim object.

Table 563. 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 564](#) details the pickmap fields for the claim object.

Table 564. Pick Map Fields for the Claim Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

[Table 565](#) provides a list of the user key combinations for the claim object.

Table 565. User Key Fields on the Claim Object

User Key Field Combinations
Id
ExternalSystemId
ClaimNumber

[Table 566](#) details the picklists available for the claim object.

Table 566. Picklists Available for the Claim Object

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

Table 566. Picklists Available for the Claim Object

Field Name
RelationshipToInsured
ReportedBy
State
Status
TypeofInjury

Coverage

The coverage object allows you to define and record details about an insurance policy coverage. Coverage is a term used to describe the monetary limits and risks covered as set out in an insurance policy.

Child Objects

None

Methods Called

[Table 567](#) details the methods called by the Coverage service.

Table 567. Methods Called by Coverage Service

Method	Name as Defined in Service
"Delete" on page 105	CoverageDelete
"Execute" on page 108	CoverageExecute
"Insert" on page 112	CoverageInsert
"QueryPage" on page 115	CoverageQueryPage
"Update" on page 136	CoverageUpdate

Fields

[Table 568](#) details the required and read-only fields for the coverage object.

Table 568. 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 569](#) details the pickmap fields for the coverage object.

Table 569. Pick Map Fields for the Coverage Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

[Table 570](#) provides a list of the user key combinations for the coverage object.

Table 570. User Key Fields on the Coverage Object

User Key Field Combinations
CoverageName
IntegrationId
ExternalSystemId

[Table 571](#) details the picklists available for the coverage object.

Table 571. Picklists Available for the Coverage Object

Field Name
Coverage
Status

Damage

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

Child Objects

None

Methods Called

[Table 572](#) details the methods called by the Damage service.

Table 572. Methods Called by Damage Service

Method	Name as Defined in Service
"Delete" on page 105	DamageDelete
"Execute" on page 108	DamageExecute
"Insert" on page 112	DamageInsert
"QueryPage" on page 115	DamageQueryPage
"Update" on page 136	DamageUpdate

Fields

[Table 573](#) details the required and read-only fields for the damage object.

Table 573. Read-Only Fields on the Damage Object

Field Name	Type
ClaimNumber	Required
DamageName	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
PrimaryContact	Read-only
Audit Fields	Read-only

Table 574 details the pickmap fields for the damage object.

Table 574. Pick Map Fields for the Damage Object

Pick Map Field	Maps To
ClaimExternalId	ClaimId
ClaimIntegrationId	ClaimId

Table 575 provides a list of the user key combinations for the damage object.

Table 575. User Key Fields on the Damage Object

User Key Field Combinations
Id
ExternalSystemId
DamageName

Financial Account

The financial account object stores information about the financial accounts of a contact or business that a financial institution manages, but can also track held away financial accounts.

Child Objects

[Activity](#), [Financial Account](#), [Financial Account Holder](#), [Financial Account Holding](#), [Financial Plan](#), [Financial Transaction](#), [Lead](#), [Contact Team](#), [Policy](#), [Service Request](#).

Methods Called

Table 576 details the methods called by the Financial Account service.

Table 576. Methods Called by Financial Account Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialAccountDelete
"Execute" on page 108	FinancialAccountExecute
"Insert" on page 112	FinancialAccountInsert
"QueryPage" on page 115	FinancialAccountQueryPage
"Update" on page 136	FinancialAccountUpdate

Fields

[Table 577](#) details the required and read-only fields for the financial account object.

Table 577. Required and Read-Only Fields for the Financial Account Object

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

[Table 578](#) details the pick map fields for the financial account object.

Table 578. Pick Map Fields for the Financial Account Object

Pick Map Field	Maps To
ParentFinancialAccountIntegrationId	ParentFinancialAccountId
ParentFinancialAccountExternalSystemId	ParentFinancialAccountId

[Table 579](#) provides a list of the user key combinations for the financial account object.

Table 579. User Key Fields on the Financial Account Object

User Key Field Combinations
FinancialAccount
Id
ExternalSystemId

[Table 580](#) details the picklists available for the financial account object.

Table 580. Picklists Available for the Financial Account Object

Field Name
Type
HomeBranch

Financial Account Holder

The financial account holder object stores information about a financial account holder, which is any contact that has a relationship with the financial account, most commonly the legal owner of the financial account.

Child Objects

None

Methods Called

[Table 581](#) details the methods called by the Financial Account Holder service.

Table 581. Methods Called by Financial Account Holder Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialAccountHolderDelete
"Execute" on page 108	FinancialAccountHolderExecute
"Insert" on page 112	FinancialAccountHolderInsert
"QueryPage" on page 115	FinancialAccountHolderQueryPage
"Update" on page 136	FinancialAccountHolderUpdate

Fields

[Table 582](#) details the required and read-only fields for the financial account holder object.

Table 582. Required and Read-Only Fields for the Financial Account Holder Object

Field Name	Type
FinancialAccountHolderName	Required
FinancialAccount	Required
Role	Required
CurrencyCode	Read-only
Audit Fields	Read-only

[Table 583](#) details the pick map fields for the financial account holder object.

Table 583. Pick Map Fields for the Financial Account Holder Object

Pick Map Field	Maps To
FinancialAccountIntegrationId	FinancialAccountId
FinancialAccountExternalSystemId	FinancialAccountId

[Table 584](#) provides a list of the user key combinations for the financial account holder object.

Table 584. User Key Fields on the Financial Account Holder Object

User Key Field Combinations
FinancialAccountHolderName
Id
ExternalSystemId

[Table 585](#) details the picklists available for the financial account holder object.

Table 585. Picklists Available for the Financial Account Holder Object

Field Name
Role

Financial Account Holding

The financial account holding object stores information about the financial account holdings, which are typically the total of all financial account transactions of a single financial product for a financial account.

Child Objects

None

Methods Called

[Table 586](#) details the methods called by the Financial Account Holding service.

Table 586. Methods Called by Financial Account Holding Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialAccountHoldingDelete
"Execute" on page 108	FinancialAccountHoldingExecute
"Insert" on page 112	FinancialAccountHoldingInsert
"QueryPage" on page 115	FinancialAccountHoldingQueryPage
"Update" on page 136	FinancialAccountHoldingUpdate

Fields

[Table 587](#) details the required and read-only fields for the financial account holding object.

Table 587. Required and Read-Only Fields for the Financial Account Holding Object

Field Name	Type
FinancialAccountHoldingsName	Required
FinancialAccount	Required
FinancialProductFinancialProductName	Required
CurrencyCode	Read-only
Audit Fields	Read-only

[Table 588](#) details the pick map fields for the financial account holding object.

Table 588. Pick Map Fields for the Financial Account Holding Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

Table 589 provides a list of the user key combinations for the financial account holding object.

Table 589. User Key Fields on the Financial Account Holding Object

User Key Field Combinations
FinancialAccountHoldingsName
Id
ExternalSystemId

Financial Plan

The financial plan object stores information about the plans of a contact or business for one or more specific financial accounts that a financial institution manages.

Child Objects

[Activity](#), [Lead](#), [Contact Team](#), [Service Request](#).

Methods Called

Table 590 details the methods called by the Financial Plan service.

Table 590. Methods Called by Financial Plan Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialPlanDelete
"Execute" on page 108	FinancialPlanExecute
"Insert" on page 112	FinancialPlanInsert
"QueryPage" on page 115	FinancialPlanQueryPage
"Update" on page 136	FinancialPlanUpdate

Fields

[Table 591](#) details the required and read-only fields for the financial plan object.

Table 591. 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 592](#) details the pick map fields for the financial plan object.

Table 592. Pick Map Fields for the Financial Plan Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

[Table 593](#) provides a list of the user key combinations for the financial plan object.

Table 593. User Key Fields on the Financial Plan Object

User Key Field Combinations
FinancialPlan
Id
ExternalSystemId

[Table 594](#) details the picklists available for the financial plan object.

Table 594. Picklists Available for the Financial Plan Object

Field Name
Type
Status

Financial Product

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

Child Objects

[Coverage](#), [Financial Account Holding](#), [Financial Product](#), [Financial Transaction](#).

Methods Called

[Table 595](#) details the methods called by the Financial Product service.

Table 595. Methods Called by Financial Product Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialProductDelete
"Execute" on page 108	FinancialProductExecute
"Insert" on page 112	FinancialProductInsert
"QueryPage" on page 115	FinancialProductQueryPage
"Update" on page 136	FinancialProductUpdate

Fields

[Table 596](#) details the required and read-only fields for the financial product object.

Table 596. Required and Read-Only Fields for the Financial Product Object

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

[Table 597](#) details the pick map fields for the financial product object.

Table 597. Pick Map Fields for the Financial Product Object

Pick Map Field	Maps To
ParentFinancialProductIntegrationId	ParentFinancialProductId
ParentFinancialProductExternalSystemId	ParentFinancialProductId

[Table 598](#) provides a list of the user key combinations for the financial product object.

Table 598. User Key Fields on the Financial Product Object

User Key Field Combinations
FinancialProductName
Id
ExternalSystemId

[Table 599](#) details the picklists available for the financial product object.

Table 599. Picklists Available for the Financial Product Object

Field Name
Category
Type
Class
Sub-Class

Financial Transaction

The financial transaction object stores information about financial account transactions, which are typically the individual transactions of a financial account for all financial products.

Child Objects

[Financial Transaction](#), [Service Request](#).

Methods Called

[Table 600](#) details the methods called by the Financial Transaction service.

Table 600. Methods Called by Financial Transaction Service

Method	Name as Defined in Service
"Delete" on page 105	FinancialTransactionDelete
"Execute" on page 108	FinancialTransactionExecute
"Insert" on page 112	FinancialTransactionInsert

Table 600. Methods Called by Financial Transaction Service

Method	Name as Defined in Service
“QueryPage” on page 115	FinancialTransactionQueryPage
“Update” on page 136	FinancialTransactionUpdate

Fields

[Table 601](#) details the required and read-only fields for the financial transaction object.

Table 601. 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 602](#) details the pick map fields for the financial transaction object.

Table 602. Pick Map Fields for the Financial Transaction Object

Pick Map Field	Maps To
FinancialTransactionParentIntegrationId	FinancialTransactionParentId
FinancialTransactionParentExternalSystemId	FinancialTransactionParentId

[Table 603](#) provides a list of the user key combinations for the financial transaction object.

Table 603. User Key Fields on the Financial Transaction Object

User Key Field Combinations
TransactionID
Id
ExternalSystemId

Table 604 details the picklists available for the financial transaction object.

Table 604. Picklists Available for the Financial Transaction Object

Field Name
TransactionType

Household

The household object allows you to define and record financial details about a group of contacts that live in the same household, for example, parents, brothers, sisters, spouses, and so on. These details include the assets of the household, the liabilities of the household, the net income of the household, and so on.

Child Objects

[Claim](#), [Financial Account](#), [Financial Account Holder](#), [Financial Plan](#), [Policy](#), [Policy Holder](#).

Methods Called

Table 605 details the methods called by the Household service.

Table 605. Methods Called by Household Service

Method	Name as Defined in Service
"Delete" on page 105	HouseholdDelete
"Execute" on page 108	HouseholdExecute
"Insert" on page 112	HouseholdInsert
"QueryPage" on page 115	HouseholdQueryPage
"Update" on page 136	HouseholdUpdate

Fields

Table 606 details the required and read-only fields for the household object.

Table 606. Required and Read-Only Fields for the Household Object

Field Name	Type
HouseholdName	Required
IntegrationID	Required
ExternalSystemID	Required
HouseholdId	Read-only

Table 606. Required and Read-Only Fields for the Household Object

Field Name	Type
PrimaryContactId	Read-only
PrimaryContactExternalId	Read-only
PrimaryContactIntegrationId	Read-only
PrimaryContactFirstName	Read-only
PrimaryContactLastName	Read-only
Timezone	Read-only
HouseholdCurrency	Read-only
LastActivity	Read-only
HeadDOB	Read-only
TotalIncome	Read-only
TotalAssets	Read-only
TotalExpenses	Read-only
TotalLiabilities	Read-only
TotalNetWorth	Read-only
RiskProfile	Read-only
ExperienceLevel	Read-only
InvestmentHorizon	Read-only
CurrentInvestmentMix	Read-only
Objective	Read-only
PrimaryGoal	Read-only
Audit Fields	Read-only

[Table 607](#) details the pick map fields for the household object.

Table 607. Pick Map Field for the Household Object

Pick Map Field	Maps To
ContactExternalId	ContactId
ContactIntegrationId	

Table 608 provides a list of the user key combinations for the household object.

Table 608. User Key Fields on the Household Object

User Key Field Combinations
HouseholdName
Id
ExternalSystemId

Table 609 details the picklists available for the household object.

Table 609. Picklists Available for the Household Object

Field Name
Segment
Type

Insurance Property

The insurance property object allows you to define and record details about an insurance property. Insurance properties are typically the properties of a contact that are included in an insurance policy or a claim.

Child Objects

[Coverage](#), [Damage](#).

Methods Called

Table 610 details the methods called by the Insurance Property service.

Table 610. Methods Called by Insurance Property Service

Method	Name as Defined in Service
"Delete" on page 105	InsurancePropertyDelete
"Execute" on page 108	InsurancePropertyExecute
"Insert" on page 112	InsurancePropertyInsert
"QueryPage" on page 115	InsurancePropertyQueryPage
"Update" on page 136	InsurancePropertyUpdate

Fields

[Table 611](#) details the required and read-only fields for the insurance property object.

Table 611. 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 612](#) details the pickmap fields for the insurance property object.

Table 612. Pick Map Fields for the Insurance Property Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

[Table 613](#) provides a list of the user key combinations for the insurance property object.

Table 613. User Key Fields on the Insurance Property Object

User Key Field Combinations
Type
Id
ExternalSystemId

[Table 614](#) details the picklists available for the insurance property object.

Table 614. Picklists Available for the Insurance Property Object

Field Name
Type

Involved Party

The involved party object allows you to define and record details about an involved party, which is typically a contact involved in an insurance claim.

Child Objects

None

Methods Called

[Table 615](#) details the methods called by the Involved Party service.

Table 615. Methods Called by Involved Party Service

Method	Name as Defined in Service
"Delete" on page 105	InvolvedPartyDelete
"Execute" on page 108	InvolvedPartyExecute
"Insert" on page 112	InvolvedPartyInsert
"QueryPage" on page 115	InvolvedPartyQueryPage
"Update" on page 136	InvolvedPartyUpdate

Fields

[Table 616](#) details the required and read-only fields for the involved party object.

Table 616. 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 617](#) details the pickmap fields for the involved party object.

Table 617. Pick Map Fields for the Involved Party Object

Pick Map Field	Maps To
ClaimExternalSystemId	ClaimId
ClaimIntegrationId	ClaimId

[Table 618](#) provides a list of the user key combinations for the involved party object.

Table 618. User Key Fields on the Involved Party Object

User Key Field Combinations
InvolvedPartyName
Id
ExternalSystemId

[Table 619](#) details the picklists available for the involved party object.

Table 619. Picklists Available for the Involved Party Object

Field Name
Role
RoleinAccident
Location

Policy

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

Child Objects

[Activity](#), [Claim](#), [Coverage](#), [Insurance Property](#), [Lead](#), [Contact Team](#), [Policy](#), [Policy Holder](#), [Service Request](#).

Methods Called

Table 620 details the methods called by the Policy service.

Table 620. Methods Called by Policy Service

Method	Name as Defined in Service
"Delete" on page 105	PolicyDelete
"Execute" on page 108	PolicyExecute
"Insert" on page 112	PolicyInsert
"QueryPage" on page 115	PolicyQueryPage
"Update" on page 136	PolicyUpdate

Fields

Table 621 details the required and read-only fields for the policy object.

Table 621. 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 622 details the pickmap fields for the policy object.

Table 622. Pick Map Fields for the Policy Object

Pick Map Field	Maps To
ClaimExternalSystemId	ClaimId
ClaimIntegrationId	ClaimId

[Table 623](#) provides a list of the user key combinations for the policy object.

Table 623. User Key Fields on the Policy Object

User Key Field Combinations
PolicyNumber
Id
ExternalSystemId

[Table 624](#) details the picklists available for the policy object.

Table 624. Picklists Available for the Policy Object

Field Name
PolicyType
Status
SubStatus
BillingStatus
RateState
RatePlan
ReferralSource
PolicyPayMethod

Policy Holder

The policy holder object allows you to define and record details about an insurance policy holder. A policy holder is typically the contact that owns the policy that the insurance company manages.

Child Objects

None

Methods Called

[Table 625](#) details the methods called by the Policy Holder service.

Table 625. Methods Called by Policy Holder Service

Method	Name as Defined in Service
"Delete" on page 105	PolicyHolderDelete
"Execute" on page 108	PolicyHolderExecute
"Insert" on page 112	PolicyHolderInsert
"QueryPage" on page 115	PolicyHolderQueryPage
"Update" on page 136	PolicyHolderUpdate

Fields

[Table 626](#) details the required and read-only fields for the policy holder object.

Table 626. 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 627](#) details the pickmap fields for the policy holder object.

Table 627. Pick Map Fields for the Policy Holder Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

[Table 628](#) provides a list of the user key combinations for the policy holder object.

Table 628. User Key Fields on the Policy Holder Object

User Key Field Combinations
PolicyHolderName
Id
ExternalSystemId

[Table 629](#) details the picklists available for the policy holder object.

Table 629. Picklists Available for the Policy Holder Object

Field Name
InsuredType
Role

Portfolio

The portfolio object allows you to define and record details about the collection of financial services that you can provide to an account. Financial services include loans, credit cards, insurance, general banking, and so on.

Child Objects

[Financial Account](#), [Financial Plan](#).

Methods Called

[Table 630](#) details the methods called by the Portfolio service.

Table 630. Methods Called by Portfolio Service

Method	Name as Defined in Service
"Delete" on page 105	PortfolioDelete
"Execute" on page 108	PortfolioExecute
"Insert" on page 112	PortfolioInsert
"QueryPage" on page 115	PortfolioQueryPage
"Update" on page 136	PortfolioUpdate

Fields

[Table 631](#) details the read-only fields for the portfolio object.

Table 631. 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 632](#) details the pickmap fields for the portfolio object.

Table 632. 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 633](#) provides a list of the user key combinations for the portfolio object.

Table 633. User Key Fields on the Portfolio Object

User Key Field Combinations
AccountNumber and ProductId
Id
ExternalSystemId

[Table 634](#) details the picklists available for the portfolio object.

Table 634. Picklists Available for the Portfolio Object

Field Name
AccountType
Status
TermUnit

Automotive Edition Parent Objects (Web Services v2.0)

This topic includes the parent objects available with Oracle CRM On Demand Automotive Edition.

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

The following objects are detailed in this topic:

- [“Dealer” on page 584](#)
- [“Vehicle” on page 586](#)

Dealer

The dealer object stores information about dealerships in the automotive industry, for example, the name of the dealership, the identity of the parent dealership, the site on which the dealership is based, and so on. The dealer object does not have any parent objects.

Child Objects

[Attachment](#), [Shared Address](#)

For information about using attachments with this object, see [Appendix A, “Using Attachments with Web Services On Demand.”](#)

Methods Called

[Table 635](#) details the methods called by the Dealer service.

Table 635. Methods Called by Dealer Service

Method	Name as Defined in Service
"Delete" on page 105	DealerDelete
"Execute" on page 108	DealerExecute
"Insert" on page 112	DealerInsert
"QueryPage" on page 115	DealerQueryPage
"Update" on page 136	DealerUpdate

Fields

[Table 636](#) details the required and read-only fields for the dealer object.

Table 636. 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 637](#) details the pick map fields for the dealer object.

Table 637. Pick Map Fields for the Dealer Object

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

Table 638 provides a list of the user key combinations for the dealer object.

Table 638. User Key Fields on the Dealer Object

User Key Field Combinations
DealerId
ExternalSystemID

Table 639 details the picklists available for the dealer object.

Table 639. Picklists Available for the Dealer Object

Field Name
ParentDealerName
ParentDealerSite

Vehicle

The vehicle object allows you to create and store information about a vehicle, for example, a car, a truck, a van, and so on, that your company would like to sell to a contact or account. This information includes the vehicle's current mileage, the invoice price, the dealership, the make, and so on.

Child Objects

None

Methods Called

Table 640 details the methods called by the Vehicle service.

Table 640. Methods Called by Vehicle Service

Method	Name as Defined in Service
"Delete" on page 105	VehicleDelete
"Execute" on page 108	VehicleExecute
"Insert" on page 112	VehicleInsert
"QueryPage" on page 115	VehicleQueryPage
"Update" on page 136	VehicleUpdate

Fields

[Table 641](#) details the required and read-only fields for the vehicle object.

Table 641. 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 642](#) details the pick map fields for the vehicle object.

Table 642. 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 643](#) provides a list of the user key combinations for the vehicle object.

Table 643. User Key Fields on the Vehicle Object

User Key Field Combinations
Id
ExternalSystemId
ProductId

Table 643. User Key Fields on the Vehicle Object

User Key Field Combinations
SerialNumber
ProductExternalId

[Table 644](#) details the picklists available for the vehicle object.

Table 644. Picklists Available for the Vehicle Object

Field Name
Body
Door
Engine
ExteriorColor
InteriorColor
Location
Make
Model
VehicleOwnedBy
Status
Transmission
Trim
UsedNew
WarrantyType
Year

Child Objects (Web Services v2.0)

This topic includes a list of child objects that are used in Oracle On Demand Web Services.

The list includes:

- Objects that are child objects only and are not themselves parent objects.
- Objects that are also parent objects.

Various parent objects are also child objects of other objects. This includes a number of industry-specific objects added in Release 16 and Release 17 of Oracle CRM On Demand. See the topics about the parent objects for information about the parent-child relationships. For these objects, the field information given for parent objects is the same for the objects when they are child objects, and is therefore not repeated in this topic.

For each child object, the methods that can be called are as shown in [Table 645](#). The methods that you can use on a child object are the same as for its parent object.

Table 645. Methods Called On Child Objects

Method	Name as Defined in Service
"Delete" on page 105	<object>Delete
"Execute" on page 108	<object>Execute
"Insert" on page 112	<object>Insert
"QueryPage" on page 115	<object>QueryPage
"Update" on page 136	<object>Update

The following child objects are listed:

- ["Account" on page 590](#)
- ["Account Contact" on page 591](#)
- ["Account Opportunity" on page 593](#)
- ["Account Team" on page 594](#)
- ["Activity" on page 596](#)
- ["Address" on page 598](#)
- ["Asset" on page 600](#)
- ["Attachment" on page 602](#)
- ["Attendee Call" on page 602](#)
- ["Book" on page 605](#)
- ["Campaign" on page 606](#)
- ["Competitor" on page 608](#)
- ["Contact" on page 609](#)
- ["Contact Role" on page 610](#)
- ["Contact Team" on page 612](#)
- ["Custom Object 1 through 3" on page 613](#)
- ["Custom Object 4 through 15" on page 614](#)
- ["Delegated User" on page 615](#)

- "Exchange Rate History" on page 616
- "Interests" on page 617
- "Invitee" on page 618
- "Lead" on page 619
- "Note" on page 621
- "Opportunity" on page 621
- "Opportunity Child" on page 622
- "Opportunity Team" on page 622
- "Products Detailed" on page 624
- "Promotional Items Dropped" on page 625
- "Recipient" on page 626
- "Related Contact" on page 627
- "Revenue" on page 628
- "Sales Assessment" on page 629
- "Sales Assessment Value" on page 630
- "Sample Dropped" on page 631
- "Service Request" on page 632
- "Shared Address" on page 634
- "User" on page 637

NOTE: For the Quote and Order child objects, access is included for use by the Oracle CRM On Demand to Oracle E-Business Suite integration and is not available for non-Oracle use. For access to the Order child object, the Order Usage value on the company profile page must be set to Ebiz PIP integration. If the Order Usage value on the company profile page is set to CRM On Demand Order Management, the Order object is a parent object with a child Order Item object.

Account

The account object stores information about the companies that you do business with and is also used to track partners and competitors.

Parent Object

[Contact](#)

Fields

[Table 646](#) details the required and read-only fields for the account child object.

Table 646. Required and Read-Only Fields for the Account Child Object

Field Name	Type
AccountId	Read-only

[Table 647](#) details the pick map fields for the account child object.

Table 647. Pick Map Fields for the Account Child Object

Pick Map Field	Maps To
ExternalSystemId	AccountId

[Table 648](#) provides a list of the user key combinations for the account child object.

Table 648. User Key Fields on the Account Child Object

User Key Field Combinations
AccountId
ExternalSystemId
Name and Location

[Table 649](#) details the picklists available for the account child object.

Table 649. Picklists Available for the Account Child Object

Field Name
Call Frequency
Route
Status
Type

Account Contact

The account contact object stores information about the contacts associated with an account.

Parent Object

[Account](#)

Fields

[Table 650](#) details the required and read-only fields for the account contact child object.

Table 650. Required and Read-Only Fields for the Account Contact Child Object

Field Name	Type
AccountLocation	Read-only
AccountMainPhone	Read-only
AccountType	Read-only

[Table 651](#) details the pick map fields for the account contact child object.

Table 651. Pick Map Fields for the Account Contact Child Object

Pick Map Field	Maps To
AccountExternalSystemId	AccountId
AccountName	AccountId
ContactExternalSystemId	ContactId
ContactFullName	ContactId
ContactLastName	ContactFirstName
ContactLastName	ContactId

[Table 652](#) provides a list of the user key combinations for the account contact child object.

Table 652. User Key Fields on the Account Contact Child Object

User Key Field Combinations
AccountId
AccountExternalSystemId
AccountLocation and AccountName
ContactExternalSystemId
ContactFirstName and ContactLastName

Table 653 details the picklists available for the account contact child object.

Table 653. Picklists Available for the Account Contact Child Object

Field Name
Role

Account Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Parent Object

[Account](#)

Fields

Table 654 details the required and read-only fields for the account opportunity child object.

Table 654. Required and Read-Only Fields for the Account Opportunity Child Object

Field Name	Type
OwnerFullName	Read-only
PrimaryRevenueExpectedValue	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
FuriganaAccountName	Read-only
NameShadow	Read-only
AssignmentStatus	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only

Table 655 details the pick map fields for the account opportunity child object.

Table 655. Pick Map Fields for the Account Opportunity Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId
AccountLocation	AccountId
Account	AccountId
SourceCampaignExternalId	CampaignId
SourceCampaign	CampaignId
KeyContactExternalId	KeyContactId
KeyContactLastName	KeyContactId
DealerExternalId	DealerId
OwnerAlias	OwnerId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 656 provides a list of the user key combinations for the account opportunity child object.

Table 656. User Key Fields on the Account Opportunity Child Object

User Key Field Combinations
OpportunityId
Id

Account Team

The team object stores information on the team that is assigned to a particular account or contact. In this way, a team of employees can be dedicated to an account or contact, ensuring that the activities, service requests, leads, and opportunities surrounding that account or contact are always kept up-to-date and are attended to regularly.

Parent Object

[Account](#)

Fields

[Table 657](#) details the required and read-only fields for the account team child object.

Table 657. Required and Read-Only Fields for the Account Team Child Object

Field Name	Type
AccountAccess	Required
UserId	Required
FirstName	Read-only

[Table 658](#) details the pick map fields for the account team child object.

Table 658. Pick Map Fields for the Account Team Child Object

Pick Map Field	Maps To
AccountAccess	AccountAccessId
AccountName	AccountId
ContactAccess	ContactAccessId
LastName	FirstName
OpportunityAccess	OpportunityAccessId
UserAlias	UserId
UserEmail	UserId
UserExternalSystemId	UserId

[Table 659](#) provides a list of the user key combinations for the account team child object.

Table 659. User Key Fields on the Account Team Child Object

User Key Field Combinations
UserId
UserExternalSystemId
UserEmail
UserAlias

[Table 660](#) details the picklists available for the account team child object.

Table 660. Picklists Available for the Account Team Child Object

Field Name
TeamRole

Activity

The activity object stores information on an activity that a user must carry out, for example, a callback activity for an account.

Parent Objects

[Account](#), [Campaign](#), [Contact](#), [Lead](#), [Opportunity](#).

Fields

[Table 661](#) details the required and read-only fields for the activity child object.

Table 661. Required and Read-Only Fields for the Activity Child Object

Field Name	Type
AccountLocation	Read-only
CreatedDetail	Read-only
MEEventName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
CODInteractionTime	Read-only
CODWrapUpTime	Read-only
CODHandleTime	Read-only
CODIVRTTime	Read-only
CODQueueHoldTime	Read-only
CODTotalHoldTime	Read-only
DescriptionShadow	Read-only
DealerName	Read-only
Audit Fields	Read-only

Table 662 details the pick map fields for the activity child object.

Table 662. Pick Map Fields for the Activity Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId
CampaignExternalId	CampaignId
CampaignIntegrationId	CampaignId
DelegatedBy	DelegatedById
FundRequest	FundRequestId
OpportunityIntegrationId	OpportunityId
OpportunityExternalId	OpportunityId
PrimaryContactIntegrationId	PrimaryContactId
PrimaryContactExternalId	PrimaryContactId
LeadIntegrationId	LeadId
LeadExternalId	LeadId
OwnerExternalId	OwnerId
OwnerIntegrationId	OwnerId
SRIntegrationId	SRId
SRExternalId	SRId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 663 provides a list of the user key combinations for the activity child object.

Table 663. User Key Fields on the Activity Child Object

User Key Field Combinations
Id
ExternalSystemId
Description

Address

The address child object stores information on the different addresses that are associated with accounts and contacts.

This child object is applicable for companies that do not have the Enable Shared Addresses check box selected on the company profile. In this case, address information is not shared between any records, and the address information in each record is specific to each record.

When Enable Shared Addresses is selected, the address child object, (<ListOfAddress>) cannot be used, and instead the shared address child object (<ListOfSharedAddress>) must be used to associate or dissociate addresses and flag addresses as billing, shipping, or primary addresses. For more information about the shared address child object, see ["Shared Address" on page 634](#). To create and update shared addresses, the top-level Address Web service must be used. For more information, see ["Address" on page 440](#).

When Enable Shared Addresses is deselected, the shared address child object cannot be used as addresses are not being shared. Instead, the address child object is used for address operations. See [Table 664](#), which shows the usage of address child objects and fields depending on whether addresses are shared. For more information about address operations, see ["Adding and Updating Account Addresses" on page 433](#), and ["Adding and Updating Contact Addresses" on page 450](#).

Table 664. Usage of Address Child Objects and Fields Depending on Whether Enable Shared Addresses is Selected

Enable Shared Addresses check box on company profile	ListOfAddress (address as child object)	ListOfSharedAddress (shared address as child object)	Denormalised address fields in parent object (PrimaryBillToCity, PrimaryShipToCity and so on)	Address Id fields in parent object (PrimaryBillToAddressId, PrimaryShipToAddressId, AlternateAddressId)
Selected - addresses are shared.	Cannot be used to create, update, or delete the addresses as addresses are being shared, but can be used for query operations.	Can be used only to associate or dissociate an address from parent. Addresses cannot be created, deleted, or updated using this child object. The top-level Address Web service must be used, see "Address" on page 440 .	Cannot be used because addresses are being shared. To create and update addresses, the top-level Address Web service must be used. see "Address" on page 440 .	Can be used to associate a billing/shipping/primary address to the account, contact, dealer, partner objects. Billing/shipping/primary fields in ListOfSharedAddress can also be used for the same purpose.
Deselected - addresses are not being shared.	Can be used to create, update, delete, and read addresses.	Partial update allowed - can be used only to edit the intersection object fields other than AddressId, billing/shipping/primary flag fields. Changing the address association or address flags is not allowed but any other fields (for example, custom fields on the account address object) can be updated. Create or delete is not allowed and change in address association is not allowed, but query operations are not restricted.	Can be used.	Usage is as described in "Adding and Updating Account Addresses" on page 433 .

Parent Objects

[Account](#), [Contact](#)

Fields

[Table 665](#) details the required and read-only fields for the address child object.

Table 665. Required and Read-Only Fields for the Address Child Object

Field Name	Type
AddressId	Read-only

[Table 666](#) details the pick map fields for the address child object.

Table 666. Pick Map Fields for the Address Child Object

Pick Map Field	Maps To
None	

[Table 667](#) provides a list of the user key combinations for the address child object.

Table 667. User Key Fields on the Address Child Object

User Key Field Combinations
AddressId
ExternalId

Asset

The asset object stores information on the assets held by your accounts, for example, the products that an account has purchased.

Parent Object

[Account](#)

Fields

[Table 668](#) details the required and read-only fields for the asset child object.

Table 668. Required and Read-Only Fields for the Asset Child Object

Field Name	Type
AssetId	Required
ContactAssetId	Read-only
ExternalSystemId	Read-only
Product	Read-only
ProductId	Read-only
ProductPartNumber	Read-only
ProductType	Read-only
ProductStatus	Read-only
CreatedByandDate	Read-only
ModifiedByandDate	Read-only
SerialNumber	Read-only

[Table 669](#) details the pick map fields for the asset child object.

Table 669. Pick Map Fields for the Asset Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
Manufacturer	ManufacturerId
PreferredServiceDealer	PreferredServiceDealerID

[Table 670](#) provides a list of the user key combinations for the asset child object.

Table 670. User Key Fields on the Asset Child Object

User Key Field Combinations
AssetId
ExternalSystemId

Attachment

The attachment object stores information about a file or URL that is attached to a record in Oracle CRM On Demand.

Parent Objects

[Account](#), [Accreditation](#), [Activity](#), [Campaign](#), [Certification](#), [Contact](#), [Course](#), [CustomObject](#), [Dealer](#), [Exam](#), [Lead](#), [Opportunity](#), [Service Request](#), and [Solution](#).

Fields

[Table 671](#) details the required and read-only fields for the attachment object.

Table 671. Required and Read-Only Fields for the Attachment Child Object

Field Name	Type
DisplayFileName	Required
FileNameOrURL	Required
FileDate	Read-only
FileSize	Read-only
<parent_object>Id	Read-only
Id	Read-only
ModId	Read-only
Audit Fields	Read-only

[Table 672](#) provides a list of the user key combinations for the attachment child object.

Table 672. User Key Fields on the Attachment Child Object

User Key Field Combinations
Id
ExternalSystemId
FileNameOrURL and FileExtension

Attendee Call

The Attendee Call object stores information about an individual call associated with an account call.

Parent Object

Activity

Fields

Table 673 details the required and read-only fields for the attendee call child object.

Table 673. Required and Read-Only Fields for the Attendee Call Child Object

Field Name	Type
Subject	Required
PrimaryContact	Required
Audit Fields	Read-only

Table 674 details the pick map fields for the attendee call child object.

Table 674. Pick Map Fields for the Attendee Call Child Object

Pick Map Field	Maps To
AddressName	AddressNameId
Address	AddressId
AddressExternalSystemId	AddressId
ApplicationCompanyName	ApplicationUID
ApplicationExternalSystemId	ApplicationCompanyName
BookName	BookId
BusinessPlanExternalSystemId	BusinessPlanPlanName
CampaignName	CampaignId
CampaignExternalId	CampaignId
ClaimExternalSystemId	ClaimNumber
CustomObjectnName	CustomObjectnId
CustomObjectnExternalSystemId	CustomObjectnId
DealerName	DealerId
DealerExternalId	DealerId
DelegatedBy	DelegatedById
DelegatedByExternalSystemId	DelegatedById
DelegatedByExternalSystemId	DelegatedByFirstName
DelegatedByExternalSystemId	DelegatedByLastName

Table 674. Pick Map Fields for the Attendee Call Child Object

Pick Map Field	Maps To
DealRegistrationName	DealRegistrationId
DealRegistrationName	DealRegistrationExternalSystemId
FinancialAccountExternalSystemId	FinancialAccountFinancialAccount
FinancialPlanExternalSystemId	FinancialPlanFinancialPlan
FundRequest	FundRequestId
Lead	LeadId
LeadIntegrationId	LeadId
LeadExternalSystemId	LeadId
LeadExternalSystemId	LeadFirstName
LeadExternalSystemId	LeadLastName
MDFRequestExternalSystemId	MDFRequestName
MEEEventName	MEEEventId
ObjectiveExternalSystemId	ObjectiveName
OpportunityName	OpportunityId
OpportunityIntegrationId	OpportunityId
OpportunityExternalId	OpportunityId
OpportunityExternalId	OpportunityName
Alias	OwnerId
PolicyExternalSystemId	PolicyNumber
PortfolioNumber	PortfolioId
PortfolioExternalId	PortfolioId
PrimaryContact	PrimaryContactId
PrimaryContactExternalId	PrimaryContactId
PrimaryContactExternalId	PrimaryContactFirstName
PrimaryContactExternalId	PrimaryContactLastName
Alias	PrimaryOwnerId
ServiceRequestNumber	ServiceRequestId
ServiceRequestExternalSystemId	ServiceRequestId
ServiceRequestExternalSystemId	ServiceRequestNumber
SPRequestExternalSystemId	SPRequestName

Table 675 provides a list of the user key combinations for the attendee call child object.

Table 675. User Key Fields on the Attendee Call Child Object

User Key Field Combinations
Id
PrimaryContactId
PrimaryContact
PrimaryContactExternalId

Table 676 provides a list of the picklists for the attendee call child object.

Table 676. Picklists for the Attendee Call Child Object

Field Name
Priority
ApprovalStatus
RejectReason
SubType
Type
CurrencyCode
CallResult
Status
ResolutionCode

Book

The book object provides a way of segmenting data according to the organizational units of your business, such as territories or products. Administrators can create book hierarchies based on how they want to organize your information, and then set up users to have the appropriate level of access to books.

Parent Object

[Accreditation](#), [Application](#), [Certification](#), [Course](#), [CustomObject](#), [Deal Registration](#), [Exam](#), [MedEd](#), [Messaging Plan](#), [Partner](#), [Solution](#), [Special Pricing Product](#).

Fields

[Table 677](#) details the required and read-only fields for the book child object.

Table 677. Required and Read-Only Fields for the Book Child Object

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

[Table 678](#) details the pick map fields for the book child object.

Table 678. Pick Map Fields for the Book Child Object

Pick Map Field	Maps To
BookName	BookId

[Table 679](#) provides a list of the user key combinations for the book child object.

Table 679. User Key Fields on the Book Child Object

User Key Field Combinations
BookId
BookName

Campaign

The campaign object stores information on the recipients of campaigns.

Parent Object

[Contact](#)

Fields

[Table 680](#) details the required and read-only fields for the campaign child object.

Table 680. Required and Read-Only Fields for the Campaign Child Object

Field Name	Type
CampaignName	Read-only
CampaignContactId	Read-only
Audit Fields	Read-only

[Table 681](#) details the pick map fields for the campaign child object.

Table 681. Pick Map Fields for the Campaign Child Object

Pick Map Field	Maps To
CampaignExternalSystemId	CampaignId
CampaignName	CampaignId

[Table 682](#) provides a list of the user key combinations for the campaign child object.

Table 682. User Key Fields on the Campaign Child Object

User Key Field Combinations
CampaignId
CampaignExternalId
CampaignName

[Table 683](#) details the picklists available for the campaign child object.

Table 683. Picklists Available for the Campaign Child Object

Field Name
DeliveryStatus
ResponseStatus

Competitor

The competitor object stores the information on competitors for your accounts and opportunities.

NOTE: This topic covers both the account competitor and opportunity competitor child objects. The Parent Object column in the tables in this topic indicates to which objects the information is applicable.

Parent Object

Account, Opportunity

Fields

Table 684 details the required and read-only fields for the competitor child object.

Table 684. Required and Read-Only Fields for the Competitor Child Object

Field Name	Type	Parent Object
CompetitorId	Required	Both
ReverseRole	Required	Both
Role	Required	Both
StartDate	Required	Both

Table 685 details the pick map fields for the competitor child object.

Table 685. Pick Map Fields for the Competitor Child Object

Pick Map Field	Maps To	Parent Object
CompetitorExternalId	CompetitorId	Both
CompetitorName	CompetitorId	Both
CompetitorFullName	CompetitorId	Both
CompetitorExternalId	AccountName	Opportunity
CompetitorName	AccountName	Opportunity

Table 686 provides a list of the user key combinations for the competitor child object.

Table 686. User Key Fields on the Competitor Child Object

User Key Field Combinations	Parent Object
Id	Both
Role & ReverseRole & CompetitorId	Both
Role & ReverseRole & CompetitorExternalId	Both
Role & ReverseRole & CompetitorName	Opportunity
Role & ReverseRole & AccountName	Opportunity

Table 687 details the picklists available for the competitor child object.

Table 687. Picklists Available for the Competitor Child Object

Field Name	Parent Object
ReverseRole	Both
Role	Both

Contact

The contact object stores information on individuals with whom your organization has a relationship. It allows the user to store information on individuals who are external to your company, but who are associated with the business process. Contacts stored in the Oracle CRM On Demand database can also be associated with an account.

Parent Object

[Activity](#), [Account](#), [Opportunity](#)

Fields

Table 688 details the required and read-only fields for the contact child object.

Table 688. Required and Read-Only Fields for the Contact Child Object

Field Name	Type
ContactId	Read-only
ContactFirstName	Read-only
ContactLastName	Read-only

Table 688. Required and Read-Only Fields for the Contact Child Object

Field Name	Type
ContactAccountId	Read-only
ContactAccountName	Read-only
ContactAccountLocation	Read-only
ContactAccountIntegrationId	Read-only
ContactAccountExternalSystemId	Read-only
Audit Fields	Read-only

[Table 689](#) details the pick map fields for the contact child object.

Table 689. Pick Map Fields for the Contact Child Object

Pick Map Field	Maps To
ContactExternalId	ContactId
ContactIntegrationId	ContactId

[Table 690](#) provides a list of the user key combinations for the contact child object.

Table 690. User Key Fields on the Contact Child Object

User Key Field Combinations
ActivityContactId
ContactExternalId

Contact Role

The contact role object stores information on the different roles that a contact can hold within an account or opportunity.

NOTE: This topic covers both the account contact role and opportunity contact role child objects. The Parent Object column in the tables in this topic indicates to which objects the information is applicable.

Parent Object

[Account](#), [Opportunity](#)

Fields

[Table 691](#) details the required and read-only fields for the contact role child object.

Table 691. Required and Read-Only Fields for the Contact Role Child Object

Field Name	Type	Parent Object
AccountId	Required	Both
ContactId	Required	Opportunity

[Table 692](#) details the pick map fields for the contact role child object.

Table 692. Pick Map Fields for the Contact Role Child Object

Pick Map Field	Maps To	Parent Object
AccountExternalSystemId	AccountId	Account
AccountName	AccountId	Account
ContactExternalSystemId	ContactId	Both
ContactLastName	ContactId	Opportunity
ContactExternalSystemId	ContactFirstName	Opportunity
ContactLastName	ContactFirstName	Opportunity
ContactExternalSystemId	ContactLastName	Opportunity

[Table 693](#) provides a list of the user key combinations for the contact role child object.

Table 693. User Key Fields on the Contact Role Child Object

User Key Field Combinations	Parent Object
Id	Account
Role, ContactId	Account
Role, ContactExternalSystemId	Account
Role, ContactFirstName, ContactLastName	Account
ContactExternalSystemId, BuyingRole	Opportunity
ContactFirstName, ContactLastName, BuyingRole	Opportunity

[Table 694](#) details the picklists available for the contact role child object.

Table 694. Picklists Available for the Contact Role Child Object

Field Name	Parent Object
Role	Account
BuyingRole	Opportunity

Contact Team

The team object stores information on the team that is assigned to a particular account or contact. In this way, a team of employees can be dedicated to an account or contact, ensuring that the activities, service requests, leads, and opportunities surrounding that account or contact are always kept up-to-date and are attended to regularly.

Parent Object

[Contact](#)

Fields

[Table 695](#) details the required and read-only fields for the contact team child object.

Table 695. Required and Read-Only Fields for the Contact Team Child Object

Field Name	Type
ContactTeamId	Read-only
UserFirstName	Read-only
UserLastName	Read-only
UserRole	Read-only

[Table 696](#) details the pick map fields for the contact team child object.

Table 696. Pick Map Fields for the Contact Team Child Object

Pick Map Field	Maps To
UserExternalSystemId	UserId
UserIntegrationId	UserId

Table 697 provides a list of the user key combinations for the contact team child object.

Table 697. User Key Fields on the Contact Team Child Object

User Key Field Combinations
None

Table 698 details the picklists available for the contact team child object.

Table 698. Picklists Available for the Contact Team Child Object

Field Name
TeamRole

Custom Object 1 through 3

The custom objects 1, 2, and 3.

Parent Objects

[Account](#), [Activity](#), [Contact](#), [Lead](#), [Opportunity](#)

Fields

Table 699 details the required and read-only fields for the custom object child object.

Table 699. Required and Read-Only Fields for the Custom Object Child Object

Field Name	Type
Name	Required
Id	Read-only

Table 700 provides a list of the user key combinations for the custom object child object.

Table 700. User Key Fields on the Custom Object Child Object

User Key Field Combinations
Id
ExternalSystemId
Name

No pickmap or picklist fields are available.

Custom Object 4 through 15

The custom objects 4 through 15.

Parent Objects

All top-level objects.

Fields

[Table 701](#) details the required and read-only fields for the custom object child object.

Table 701. Required and Read-Only Fields for the Custom Object Child Object

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

[Table 702](#) details the pick map fields for the custom object child object.

Table 702. Pick Map Fields for the Custom Object Child Object

Pick Map Field	Maps To
Owner	OwnerId

[Table 703](#) provides a list of the user key combinations for the custom object child object.

Table 703. User Key Fields on the Custom Object Child Object

User Key Field Combinations
Id
ExternalSystemId

Table 704 details the picklists available for the custom object child object.

Table 704. Picklists Available for the Custom Object Child Object

Field Name
Type

Delegated User

The delegated user object stores information about a user who can impersonate another user. The delegated user is added so that he or she can access a user's data, for example, when a person is on vacation or leaves the company.

Parent Object

User

Fields

Table 705 details the required and read-only fields for the delegated user child object.

Table 705. Required and Read-Only Fields for the Delegated User Child Object

Field Name	Type
UserId	Required
UserRole	Required
Audit Fields	Read-only

Table 706 provides a list of the user key combinations for the delegated user child object.

Table 706. User Key Fields on the Delegated User Child Object

User Key Field Combinations
Id
ExternalSystemId

Table 707 details the picklists available for the delegated user child object.

Table 707. Picklists Available for the Delegated User Child Object

Field Name
UserRole

Exchange Rate History

The Exchange rate history object stores historical information about exchange rates for the currencies used by a company.

Parent Object

[Exchange Rate](#)

Usage

The ExchangeRateHistory child object can be queried to give the history of exchange rates for the company.

Fields

Table 708 details the required and read-only fields for the exchange rate object.

Table 708. Required and Read-Only Fields for the Exchange Rate History Object

Field Name	Type
FromCurrencyCode	Required
ToCurrencyCode	Required
ExchangeDate	Required
ExchangeRate	Required

Table 709 provides a list of the user key combinations for the exchange rate history object.

Table 709. User Key Fields on the Exchange Rate History Object

User Key Field Combinations
FromCurrencyCode
ToCurrencyCode
ExchangeDate

Interests

The interests object stores information about things in which a contact is interested, such as products, services, or hobbies.

Parent Object

[Contact](#)

Fields

[Table 710](#) details the required and read-only fields for the interests child object.

Table 710. Required and Read-Only Fields for the Interests Child Object

Field Name	Type
Category	Required
Interests	Required
InterestId	Read-only
Audit Fields	Read-only

[Table 711](#) details the pick map fields for the interests child object.

Table 711. Pick Map Fields for the Interests Child Object

Pick Map Field	Maps To
InterestExternalSystemId	InterestId

[Table 712](#) provides a list of the user key combinations for the interests child object.

Table 712. User Key Fields on the Interests Child Object

User Key Field Combinations
InterestId
ExternalSystemId

[Table 713](#) details the picklists available for the interests child object.

Table 713. Picklists Available for the Interests Child Object

Field Name
Category
Subjects

Invitee

The invitee object stores information about invitees to medical education events, including feedback about the invitation.

Parent Object

[MedEd](#)

Fields

[Table 714](#) details the required and read-only fields for the invitee child object.

Table 714. Required and Read-Only Fields for the Invitee Child Object

Field Name	Type
InviteeStatus	Required
InviteeId	Read-only
Audit Fields	Read-only

[Table 715](#) details the pick map fields for the invitee child object.

Table 715. Pick Map Fields for the Invitee Child Object

Pick Map Field	Maps To
InviteeExternalSystemId	InviteeId

Table 716 provides a list of the user key combinations for the invitee child object.

Table 716. User Key Fields on the Invitee Child Object

User Key Field Combinations
ContactIdExternalId
MedEdInviteeId

Table 717 details the picklists available for the invitee child object.

Table 717. Picklists Available for the Invitee Child Object

Field Name
InviteeStatus

Lead

The lead object stores information on a company or individual with whom an opportunity can be created. It allows the user to identify the companies that might be interested in a product or service. Leads are usually generated as part of a marketing campaign.

Parent Objects

[Account](#), [Campaign](#), [Contact](#), [Opportunity](#)

Fields

Table 718 details the required and read-only fields for the lead child object.

Table 718. Required and Read-Only Fields for the Lead Child Object

Field Name	Type
OwnerFullName	Read-only
SalesRepFullName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
ReferredByFullName	Read-only
FuriganaAccountName	Read-only
FuriganaContactFirstName	Read-only
FuriganaContactLastName	Read-only

Table 718. Required and Read-Only Fields for the Lead Child Object

Field Name	Type
FirstNameShadow	Read-only
LastNameShadow	Read-only
CompanyNameShadow	Read-only
AssignmentStatus	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only

Table 719 details the pick map fields for the lead child object.

Table 719. Pick Map Fields for the Lead Child Object

Pick Map Field	Maps To
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

Table 720 provides a list of the user key combinations for the lead child object.

Table 720. User Key Fields on the Lead Child Object

User Key Field Combinations
LeadId
Id

Note

The note object stores information about the notes available in the Message Center in the Oracle CRM On Demand application.

Parent Objects

[Account](#), [Campaign](#), [Contact](#), [Opportunity](#), [Service Allotment](#)

Fields

Table 721 details the required and read-only fields for the note child object.

Table 721. Required and Read-Only Fields for the Note Child Object

Field Name	Type
Subject	Required
NotId	Read-only
Audit Fields	Read-only

Table 722 provides a list of the user key combinations for the note child object.

Table 722. User Key Fields on the Note Child Object

User Key Field Combinations
NotId

Opportunity

The opportunity object allows employees to identify and record a potential revenue-generating event that has arisen with an account or contact. Opportunities can be generated from marketing campaigns when leads indicate that they are interested in a product or service that has been offered.

Parent Object

[Contact](#)

Fields

[Table 723](#) details the required and read-only fields for the opportunity child object.

Table 723. Required and Read-Only Fields for the Opportunity Child Object

Field Name	Type
OpportunityName	Required
CloseDate	Required

[Table 724](#) provides a list of the user key combinations for the opportunity child object.

Table 724. User Key Fields on the Opportunity Child Object

User Key Field Combinations
Id
ExternalSystemId
OpportunityName

Opportunity Child

The opportunity child object is the child of a parent opportunity in an opportunity hierarchy. Such hierarchies are used for sales opportunities that have multiple components.

The same fields are accessible for the opportunity child object as for the opportunity parent object.

For more information about opportunity hierarchies, see *Oracle CRM On Demand Online Help*.

Parent Object

[Opportunity](#)

Opportunity Team

The opportunity team object stores information on the team that is assigned to a particular opportunity.

Parent Object

[Opportunity](#)

Fields

[Table 725](#) details the required and read-only fields for the opportunity team child object.

Table 725. Required and Read-Only Fields for the Opportunity Team Child Object

Field Name	Type
OpportunityAccess	Required
UserId	Required
UserFirstName	Read-only

[Table 726](#) details the pick map fields for the opportunity team child object.

Table 726. Pick Map Fields for the Opportunity Team Child Object

Pick Map Field	Maps To
UserLastName	UserFirstName
OpportunityAccess	OpportunityAccessId
UserAlias	UserId
UserEmail	UserId
UserExternalSystemId	UserId

[Table 727](#) provides a list of the user key combinations for the opportunity team child object.

Table 727. User Key Fields on the Opportunity Team Child Object

User Key Field Combinations
UserId
UserExternalSystemId
UserEmail
UserAlias

[Table 728](#) details the picklists available for the opportunity team child object.

Table 728. Picklists Available for the Opportunity Team Child Object

Field Name
TeamRole

Products Detailed

The Products Detailed object stores the information on product details for an activity. This is used, for example, to record information about products discussed on sales calls to customers.

Parent Object

[Activity](#)

Fields

[Table 729](#) details the required and read-only fields for the products detailed child object.

Table 729. Required and Read-Only Fields for the Products Detailed Child Object

Field Name	Type
ProductId	Required
Indication	Required
ProductDetailedId	Read-only
Audit Fields	Read-only

[Table 730](#) details the pick map fields for the products detailed child object.

Table 730. Pick Map Fields for the Products Detailed Child Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

Table 731 provides a list of the user key combinations for the products detailed child object.

Table 731. User Key Fields on the Products Detailed Child Object

User Key Field Combinations
ProductDetailedId
ProductDetailedExternalSytemId
ProductExternalSytemId
ProductId
ProductExternalSytemId
Name
Indication

Promotional Items Dropped

The Promotional Items Dropped object stores the information on promotional items dropped for an activity.

Parent Object

[Activity](#)

Fields

Table 732 details the required and read-only fields for the promotional items dropped child object.

Table 732. Required and Read-Only Fields for the Promotional Items Dropped Child Object

Field Name	Type
ProductName	Required
Quantity	Required
ProductAllocationId	Required
Audit Fields	Read-only

[Table 733](#) details the pick map fields for the promotional items dropped child object.

Table 733. Pick Map Fields for the Promotional Items Dropped Child Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

[Table 734](#) provides a list of the user key combinations for the promotional items dropped child object.

Table 734. User Key Fields on the Promotional Items Dropped Child Object

User Key Field Combinations
Id

Recipient

The recipient object stores information about a recipient associated with a campaign.

Parent Object

[Campaign](#)

Fields

[Table 735](#) details the required and read-only fields for the recipient child object.

Table 735. Required and Read-Only Fields for the Recipient Child Object

Field Name	Type
ContactId	Required
ModifiedDate	Read-only

[Table 736](#) provides a list of the user key combinations for the recipient child object.

Table 736. User Key Fields on the Recipient Child Object

User Key Field Combinations
None

Related Contact

The related contact object stores information about a contact that has a relationship with the parent contact in question. The details of the related contact child object are inherited from a particular contact parent object.

Parent Object

Contact

Fields

Table 737 details the required and read-only fields for the related contact child object.

Table 737. Required and Read-Only Fields for the Related Contact Child Object

Field Name	Type
ContactRelationshipId	Read-only
ContactId	Read-only
RelatedContactFirstName	Read-only
RelatedContactLastName	Read-only

Table 738 details the pick map fields for the related contact child object.

Table 738. Pick Map Fields for the Related Contact Child Object

Pick Map Field	Maps To
RelatedContactExternalId	RelatedContactId
RelatedContactIntegrationId	RelatedContactId

Table 739 provides a list of the user key combinations for the related contact child object.

Table 739. User Key Fields on the Related Contact Child Object

User Key Field Combinations
None

Table 740 details the picklists available for the related contact child object.

Table 740. Picklists Available for the Related Contact Child Object

Field Name
Relationship
Status

Revenue

The revenue object stores monetary information about accounts, contacts, and their associated opportunities. This includes information on the revenue available, expected revenue, and also information about the products associated with the accounts, contacts, opportunities, and so on.

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

Parent Objects

[Account](#), [Contact](#), [Opportunity](#)

Fields

Table 741 details the required and read-only fields for the revenue child object.

Table 741. Required and Read-Only Fields for the Revenue Child Object

Field Name	Type
RevenueId	Required
PartNumber	Required
Revenue	Required
ContactFullName	Required
Audit Fields	Read-only

Table 742 details the pick map fields for the revenue child object.

Table 742. Pick Map Fields for the Revenue Child Object

Pick Map Field	Maps To
Product	ProductId
ProductExternalId	ProductId
ProductIntegrationId	ProductId

Table 742. Pick Map Fields for the Revenue Child Object

Pick Map Field	Maps To
ProductCategory	ProductCategoryId
ProductCategoryExternalId	ProductCategoryId
ProductCategoryIntegrationId	ProductCategoryId

[Table 743](#) provides a list of the user key combinations for the revenue child object.

Table 743. User Key Fields on the Revenue Child Object

User Key Field Combinations
RevenueId
ProductId
ProductExternalId

[Table 744](#) details the picklists available for the revenue child object.

Table 744. Picklists Available for the Revenue Child Object

Field Name
Type
Status
Frequency

Sales Assessment

The sales assessment object stores information about a sales assessment script associated with an account, activity, business plan, or objective.

Parent Object

[Account](#), [Activity](#), [Business Plan](#), [Objective](#)

Fields

[Table 745](#) details the required and read-only fields for the sales assessment child object.

NOTE: Only one of the fields `TemplateId` or `Name` is required.

Table 745. Required and Read-Only Fields for the Sales Assessment Child Object

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

[Table 746](#) provides a list of the user key combinations for the sales assessment child object.

Table 746. User Key Fields on the Sales Assessment Child Object

User Key Field Combinations
Name
TemplateId

Sales Assessment Value

The sales assessment value object stores information about the responses for a sales assessment script associated with the parent object, that is, the values related to the sales assessment child object.

Parent Object

[Account](#), [Activity](#), [Business Plan](#), [Objective](#)

Fields

[Table 747](#) details the required and read-only fields for the sales assessment value child object.

Table 747. Required and Read-Only Fields for the Sales Assessment Value Child Object

Field Name	Type
AssessId	Required
AttribId	Required
Audit Fields	Read-only

Table 748 provides a list of the user key combinations for the sales assessment value child object.

Table 748. User Key Fields on the Sales Assessment Value Child Object

User Key Field Combinations
AssessId
AttribId

Sample Dropped

The SampleDropped object stores the information on samples for an activity. This is used, for example, to record information about samples left with the customer on sales calls to customers.

Parent Object

[Activity](#)

Fields

Table 749 details the required and read-only fields for the sample dropped child object.

Table 749. Required and Read-Only Fields for the Sample Dropped Child Object

Field Name	Type
ProductId	Required
Quantity	Required
SampleDroppedId	Read-only
Audit Fields	Read-only

Table 750 details the pick map fields for the sample dropped child object.

Table 750. Pick Map Fields for the Sample Dropped Child Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

Table 751 provides a list of the user key combinations for the sample dropped child object.

Table 751. User Key Fields on the Sample Dropped Child Object

User Key Field Combinations
SampleDroppedId
SampleDroppedExternalSystemId
ProductId
ProductExternalSystemId
Quantity
ProductName

Service Request

The service request object allows customers to request information or assistance with a problem related to products or services purchased from your company. Service requests can be ranked for severity and prioritized accordingly.

Parent Objects

[Account](#), [Contact](#)

Fields

Table 752 details the required and read-only fields for the service request child object.

Table 752. Required and Read-Only Fields for the Service Request Child Object

Field Name	Type
Name	Read-only
OwnerFullName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
LastAssessmentDate	Read-only
AssignmentStatus	Read-only
FuriganaAccountName	Read-only
FuriganaContactFirstName	Read-only
FuriganaContactLastName	Read-only

Table 752. Required and Read-Only Fields for the Service Request Child Object

Field Name	Type
SRNumberShadow	Read-only
AbstractShadow	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only
Audit Fields	Read-only

[Table 753](#) details the pick map fields for the service request child object.

Table 753. Pick Map Fields for the Service Request Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId
AccountLocation	AccountId
Account	AccountId
AssetExternalId	AssetId
Dealer	DealerId
ContactExternalId	ContactId
ContactIntegrationId	ContactId
DealerExternalId	DealerId
OwnerExternalId	OwnerId
OwnerIntegrationId	OwnerId
OwnerAlias	OwnerId
Product	ProductId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 754 provides a list of the user key combinations for the service request child object.

Table 754. User Key Fields on the Service Request Child Object

User Key Field Combinations
ServiceRequestId
Id

Shared Address

The shared address child object stores information on the addresses that can be shared between accounts, contacts, partners, and dealers.

This child object is applicable only for companies that have the Enable Shared Addresses check box selected on the company profile. In this case, addresses can be associated with multiple account, contact, dealer, or partner records.

When a child shared address is inserted, it can be flagged as a billing address or shipping address for accounts, dealers, and partners, or as a primary contact address for contacts. When this occurs, the billing, shipping, or primary contact fields in the parent object are populated from the shared address fields.

You cannot delete a shared address child object directly, it can only be dissociated through the parent-level shared address object. For more information, see ["Address" on page 440](#). For more information about the shared address functionality, see *Oracle CRM On Demand Online Help*.

When Enable Shared Addresses is deselected, the shared address child object cannot be used, because addresses are not being shared. Instead, the address child object is used for address operations. For information about the address child object, see ["Address" on page 598](#). See [Table 755](#) for information about the usage of address child objects and fields depending on whether addresses are shared.

Table 755. Usage of Address Child Objects and Fields Depending on Whether Enable Shared Addresses is Selected

Enable Shared Addresses check box on company profile	ListOfAddress (address as child object)	ListOfSharedAddress (shared address as child object)	Denormalised address fields in parent object (PrimaryBillToCity, PrimaryShipToCity and so on)	Address Id fields in parent object (PrimaryBillToAddressId, PrimaryShipToAddressId, AlternateAddressId)
Selected - addresses are shared.	Cannot be used to create, update, or delete the addresses as addresses are being shared, but can be used for query operations.	Can be used only to associate or dissociate an address from parent. Addresses cannot be created, deleted, or updated using this child object. The top-level Address Web service must be used, see "Address" on page 440 .	Cannot be used because addresses are being shared. To create and update addresses, the top-level Address Web service must be used. see "Address" on page 440 .	Can be used to associate a billing/shipping/primary address to the account, contact, dealer, partner objects. Billing/shipping/primary fields in ListOfSharedAddress can also be used for the same purpose.
Deselected - addresses are not being shared.	Can be used to create, update, delete, and read addresses.	Partial update allowed - can be used only to edit the intersection object fields other than AddressId, billing/shipping/primary flag fields. Changing the address association or address flags is not allowed but any other fields (for example, custom fields on the account address object) can be updated. Create or delete is not allowed and change in address association is not allowed, but query operations are not restricted.	Can be used.	Usage is as described in "Adding and Updating Account Addresses" on page 433 .

Parent Objects

[Account](#), [Contact](#), [Dealer](#), [Partner](#)

Fields

[Table 756](#) details the required and read-only fields for the shared address child object.

Table 756. Required and Read-Only Fields for the Shared Address Child Object

Field Name	Type
AddressId	Read-only

[Table 757](#) details the pick map fields for the shared address child object.

Table 757. Pick Map Fields for the Shared Address Child Object

Pick Map Field	Maps To
AddressName	AddressId
AddressExternalSystemId	AddressId

[Table 758](#) provides a list of the user key combinations for the shared address child object.

Table 758. User Key Fields on the Shared Address Child Object

User Key Field Combinations
Id
AddressId
AddressName
AddressExternalSystemId

[Table 759](#) details the picklists available for the shared address child object.

Table 759. Picklists Available for the Shared Address Child Object

Field Name
AddressType
CurrencyCode

User

The user object allows you to define and record details of all users in Oracle CRM On Demand, for example, name, position, contact details, manager, and so on.

Parent Object

[Activity](#)

Fields

[Table 760](#) details the required and read-only fields for the user child object.

Table 760. Required and Read-Only Fields for the User Child Object

Field Name	Type
UserId	Read-only
UserEmail	Read-only
UserFirstName	Read-only
UserLastName	Read-only
UserRole	Read-only
Audit Fields	Read-only

[Table 761](#) details the pick map fields for the user child object.

Table 761. Pick Map Fields for the User Child Object

Pick Map Field	Maps To
UserExternalId	UserId
UserIntegrationId	UserId

[Table 762](#) provides a list of the user key combinations for the user child object.

Table 762. User Key Fields on the User Child Object

User Key Field Combinations
UserId
UserExternalId

A

Using Attachments with Web Services On Demand

This appendix describes how to exchange attachments using Web services On Demand. It contains the following topics:

- [About Attachments on page 639](#)
- [The Attachment Element on page 639](#)
- [Specifying File Attachment Content on page 641](#)
- [Retrieving or Querying File Attachments on page 645](#)
- [Specifying URL Attachments on page 647](#)

About Attachments

Attachments can be exchanged for a number of objects using Web Services On Demand. The parent objects that have Attachment child objects are: Account, Accreditation, Activity, Campaign, Contact, Course, Custom Object 01 - 03, Dealer, Exam, Lead, Opportunity, Service Request, and Solution.

File Attachments can be exchanged as text or as binary data. In the case of binary data, the data must be encoded as inline Base64 encoded data, which ensures that the binary data is in a printable format. Binary files must therefore be encoded into Base64 format in the content provided in SOAP requests.

File attachments added through Web services requests are limited to 20MB in size just as they are for the Oracle CRM On Demand UI. Attachments greater than 20MB in size are therefore rejected. To accommodate Base64 encoding of binary files, the request size limit for SOAP requests with attachment files is 28800 KB.

You can develop client applications that allow attachment files to be added or updated for multiple records. SOAP requests contain multiple records; the first record contains the attachment as well as a Content ID value. However, other records in the SOAP request do not need to contain the same attachment, they need only reference the Content ID value from the first record.

The Attachment Element

In a Web services call, the content of an attachment must be contained in the <Attachment> element, which has the complex data type SiebelXmlAttachmentType, as shown in the following sample WSDL for an Attachment child object:

```
<xsd:complexType name="ListOfAttachment">
  <xsd:sequence>
    <xsd:element name="Attachment" maxOccurs="unbounded" minOccurs="0"
      type="xsdLocal1:Attachment" />
  </xsd:sequence>
</xsd:complexType>
```

```

</xsd: sequence>

</xsd: complexType>

<xsd: complexType name="Attachment">

<xsd: sequence>

    <xsd: element name="Id" maxOccurs="1" minOccurs="0" type="xsdLocal 1: string30"/>

    <xsd: element name="DisplayFileName" maxOccurs="1" minOccurs="0"
type="xsdLocal 1: string200" />

    <xsd: element name="FileNameOrURL" maxOccurs="1" minOccurs="0"
type="xsdLocal 1: string200" />

    <xsd: element name="FileExtension" maxOccurs="1" minOccurs="0"
type="xsdLocal 1: 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="xsdLocal 1: 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="xsdLocal 1: 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="xsdLocal 1: Siebel Xml AttachmentType"/>

</xsd: sequence>

</xsd: complexType>

...

<xsd: complexType name="Siebel Xml AttachmentType">

<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 643](#).
- **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">OM8R4KGxGuEAAAAAAAAAAAAAAAAAAAAA
.....
.....
            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 nonbinary 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 must only 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 reuse test</Description>

          <Location>Bangalore</Location>

          <AccountName>Attach Test Reuse</AccountName>

          <NumberOfEmployees>300</NumberOfEmployees>

          <ListOfAttachment>

            <Attachment>

              <FileNameOrURL>Attached</FileNameOrURL>

              <FileExtension>doc</FileExtension>

              <DisplayFileName>To be reused</DisplayFileName>

              <Attachment ContentId="reuse">OM8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/
COAGAAAAAAAAAAAAACAAAA+AAAAAAAAAAEAAA+gAAAAEAAAD+////AAAAAPYAAD3AAAA/////////
//////////

              ..... The Base64 encoded stream goes here .....

              AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=</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. You must only specify the ContentId attribute 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 xmlns:ns="urn: crmondemand/ws/account/10/2004">
  <ns: LastPage>true</ns: LastPage>
  <ListOfAccount xmlns="urn: /crmondemand/xml /account">
    <Account>
      <AccountId>10A2-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="10A2-RTX3">PD94bWwgdI VVRGLTgi Pz48P1NpZWJI bC1Qcm9wZXJ0eS1TZXQgRXNj YXBI
          <Base 64 encoded attachment data .....>
        </Attachment>
      </Attachment>
    </ListOfAttachment>
  </Account>
</ListOfAccount>
</ns: AccountWS_AccountQueryPage_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

There are two additional attributes of the <Attachment> element:

- **EndOfData**. Indicates that the data is the complete content. Always true.
- **TimedOut**. Indicates whether a time-out occurred when obtaining the data. Always false.

Specifying URL Attachments

Web Services also supports the exchange of URL attachments as well as file attachments.

The handling of URL attachments differs from file attachments in the following ways:

- The FileNameOrURL field must specify the actual URL. The URL must begin with one of the following: HTTP, HTTPS, FTP or WWW (case-insensitive).
- The FileNameOrURL and FileExtension fields form a user key for Attachment child objects, however, FileExtension is not a required field for URL attachments and is ignored.

The <Attachment> element is not required to have any content, and any content is ignored if provided.

The following example SOAP request shows a URL attachment.

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

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

  <soap:Body>

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

      <ListOfAccount>

        <Account>

          <Description>Account with URL attachment</Description>

          <Location>Bangalore</Location>

          <AccountName>Atach URL</AccountName>

          <NumberEmployees>300</NumberEmployees>

          <ListOfAttachment>

            <Attachment>

              <FileNameOrURL>www.pingpong.com</FileNameOrURL>

            </Attachment>

          </ListOfAttachment>

        </Account>

      </ListOfAccount>

    </AccountWS_AccountInsert_Input>

  </soap:Body>

</soap:Envelope>
```


B

Mapping of Display Names to Service API Object Names

This appendix contains the following topic:

- [“Mapping of Display Names”](#)

Mapping of Display Names

A number of the Service API methods, such as `GetMapping` use elements such as `<ObjectName>` or `<RecordType>` to identify the objects on which they operate. However, the object names specified in requests or returned might differ from the name displayed in the UI. For such cases, [Table 763](#) shows the mapping between the default UI display name of objects and their programmatic names as used by Service API methods. For more information about the Service API methods, see [“Service API Calls”](#) on page 139.

Table 763. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Account Address	AccountAddress
Account Competitor	AccountCompetitor
Account Partner	AccountPartner
Account Relationship	AccountRelationship
Accreditation Request	AccreditationRequest
Address	CUT Address
Address	SharedAddress
Allocation	CRMOD_LS_Allocation
Assessment	Sales Assessment Value
Business Plan	CRMODLS_BusinessPlan
Call Product Detail	Call ProdDetail
Call Promotional Item	Call PromItemDrop
Call Sample Dropped	Call SampDrop
Campaign Recipient	ContactCampaign
Certification Request	CertificationRequest
Contact Address	ContactAddress
Contact Best Times	ContactBestTimes

Table 763. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Contact Interests	ContactInterest
Contact Relationship	ContactRelationship
Contact State License	CRMOD_LS_ContactLicenses
Course Enrollment	CourseEnrollment
Custom Object 01 Team	CustObj1 Team
Custom Object 02 Team	CustObj2 Team
Custom Object 03 Team	CustObj3 Team
Dealer	Channel Partner
Dealer Address	DealerAddress
Dealer Territory	Channel Partner Territory
Deal Registration	DealRegistration
Deal Registration Product Revenue	DealRegistrationProductRevenue
Deal Registration Team	DealRegistrationTeam
Exam Registration	ExamRegistration
Financial Account	FinancialAccount
Financial Account Holder	FinancialAccountHolder
Financial Account Holding	FinancialAccountHolding
Financial Plan	FinancialPlan
Financial Product	FinancialProduct
Financial Transaction	FinancialTransaction
Household Team	HouseholdTeam
Inventory Audit Report	CRMODLS_InventoryAuditReport
Inventory Period	CRMODLS_InventoryPeriod
Insurance Property	InsuranceProperty
Involved Party	InvolvedParty
MDF Request	MDFRequest
MDF Request Team	MDFRequestTeam
MedEd Event	MedEdEvent
MedEd Invitee	MedEdInvitee
Message Response	CRMODLS_PCD_MSGRSP
Messaging Plan	CRMOD_LS_MessagingPlan

Table 763. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Messaging Plan Item	CRMOD_LS_MsgPlanItem
Messaging Plan Item Relation	CRMOD_LS_MsgPlnRel
Modification Tracking	CRMODLS_ModificationLog
Objective	CRMODLS_OBJECTIVE
Opportunity Competitor	OpportunityCompetitor
Opportunity Partner	OpportunityPartner
Partner Account	PartnerAccount
Partner Address	PartnerAddress
Partner Opportunity	PartnerOpportunity
Partner Program	PartnerProgram
Partner Type	PartnerType
Plan Account	CRMODLS_BPL_ACNT
Plan Contact	CRMODLS_BPL_CNTCT
Plan Opportunity	CRMODLS_PlanOpportunities
Policy Holder	Policy
Portfolio Team	PortfolioTeam
Portfolio Owner	PortfolioOwners
Price List	PriceList
Price List Line Item	PriceListLineItem
Program Membership	ProgramPartner
Sample Disclaimer	CRMODLS_SIGNDISC
Sample Inventory	CRMODLS_SampleInventory
Sample Lot	CRMODLS_SampleLot
Sample Transaction	CRMOD_LS_Transactions
Signature	CRMODLS_Signature
Special Pricing Request	SPRequest
Special Pricing Request Line Item	SPRequestLineItem
Special Pricing Request Team	SPRequestTeam
Transaction Item	CRMOD_LS_TransactionItems
Vehicle Financial Information	Vehicle FinInfo

Table 763. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Vehicle Sales History	Vehicle SalesHist
Vehicle Service History	Vehicle ServHist



Web Services On Demand Limiters

This appendix contains the following topics:

- [Web Services On Demand Limiters](#)
- [Web Services Request Rate on page 654](#)

Web Services On Demand Limiters

The Oracle CRM On Demand's Web Services interface resources can be shared by multiple organizations. Oracle CRM On Demand provides a limiting infrastructure to make sure that some users do not consume a disproportionate share of those resources. These limiters constrain customer organizations' use of server-side resources to equitably share available resources among users and to minimize the possibility of denial-of-service incidents.

[Table 764](#) summarizes the limiters, their default values, units, and error messages. For more information, contact Customer Care.

In addition to the limiters described in this appendix, there are a number of service allotments that promote the equitable use of resources. For more information, see ["About Service Allotments" on page 45](#). Further information about the Web Services Request Rate limiter is provided in ["Web Services Request Rate" on page 654](#).

Table 764. Web Services On Demand Limiters

Limiter Name	Description	Default Value	Units	Error Message
Session limit quota	Maximum number of concurrent sessions for a company	10	sessions	Not applicable
Lifetime of Audit records	Maximum number of days Web services utilization records are saved. Records older than 30 days are purged.	30	days	Not applicable
Web Service Request Rate	Number of Web service messages that can be sent in one second	20	requests /second	The maximum rate of requests was exceeded. Please try again in {0} ms.

Table 764. Web Services On Demand Limiters

Limiter Name	Description	Default Value	Units	Error Message
Web Service Request Size	Maximum size, in kilobytes, of a Web services request.	28800	KB	Request exceeded the size limit of {0} KB.
Web Service Session Inactivity Time Limit	How long, in seconds, that a Web services session can remain idle before being closed.	600	seconds	Internal Error: Session is not available. Aborting.
Maximum Records Returned	For return messages the maximum number of records returned for each query. A response never returns more than the specified number of records for a parent object in a request.	100 (fixed maximum value)	records	PageSize method argument cannot be greater than 100, specified by the server parameter 'MaximumPageSize'.
Maximum Objects in a Web Services Request	The maximum number of objects that can be sent in a single SOAP request.	20 (fixed maximum value)	objects	Requests containing more than 20 objects are not allowed.

Web Services Request Rate

All integration requests (data exchange requests) in a session are subject to rate limiting. Rate limiting is implemented for the following reasons:

- A user can perform long-running operations on the server that result in complex and long-running queries on the database.
- A user can perform constant operations on the server that constantly use resources.

Rate limiting can alleviate the previous problems to some extent. Oracle CRM On Demand applies a restriction to each session to limit the number of requests for each second that clients can make. The rate limit is set to twenty requests for each second. This is measured as a minimum of 1/20th second wait time between requests.

If the rate limit is exceeded, the following error message is provided to subsequent SOAP requests:

The maximum rate of requests was exceeded. Please try again in <waitTime> ms.

There are two situations in which this error message is displayed:

- Where the client sends more than 20 requests every second

- Where the client sends requests at a rate faster than one request every 50 milliseconds. For the second situation, a wait statement is recommended.

Index

A

AccessProfileRead method

arguments 214
call return value 214

AccessProfileReadAll method

arguments 215
call return value 215

AccessProfileUpsert method

arguments 216
call return value 216

account child object

about 590
fields 591
pick map 591
picklists 591
read-only fields 591
required fields 591
user key fields 591

account contact child object

about 591
fields 592
pick map 592
picklists 593
read-only fields 592
required fields 592
user key fields 592

account contact role child object

user key fields 611

account object

about 291, 433
child components 291
fields 292, 434
filterable fields 303
methods 292, 434
pick map fields 298, 434
picklists 309, 435
read-only fields 292, 434
required fields 292, 434
status key 296
user key fields 435

account team child object

about 594
fields 595
pick map 595
picklists 596
read-only fields 595
required fields 595

user key fields 595

accreditation object

about 476
fields 477
list of values 478
methods 477
pick map fields 477
read-only fields 477
required fields 477
user key fields 478

accreditation request object

about 478
fields 479
list of values 480
methods 479
pick map fields 479
read-only fields 479
required fields 479
user key fields 480

ActionBarCustomWebAppletRead method

arguments 217
call return value 217

ActionBarCustomWebAppletReadAll method

arguments 217
call return value 217

ActionBarCustomWebAppletUpsert method

arguments 219
call return value 219

ActionBarLayoutReadAll method

arguments 219
call return value 219

ActionBarLayoutUpsert method

arguments 221
call return value 221

activity child object

about 596
fields 596
pick map 597
read-only fields 596
required fields 596
user key fields 597

activity object

about 310, 435
child components 310
fields 312, 437
filterable fields 316
list of values 318, 439

- methods 311, 437
- parents 310
- pick map fields 315, 438
- read-only fields 312, 437
- required fields 312, 437
- status key 314
- usage 310, 400, 435, 474
- user key fields 316, 439
- address child object** 598
 - fields 600
 - pick map 600
 - read-only fields 600
 - required fields 600
 - user key fields 600
- address fields** 35
- Administrative Services API calls**
 - AccessProfileRead 214
 - AccessProfileReadAll 215
 - AccessProfileUpsert 215
 - ActionBarCustomWebAppletRead 216
 - ActionBarCustomWebAppletReadAll 217
 - ActionBarCustomWebAppletUpsert 218
 - ActionBarLayoutReadAll 219
 - ActionBarLayoutUpsert 220
 - CascadingPicklistRead 221
 - CascadingPicklistReadAll 222
 - CascadingPicklistUpsert 223
 - CurrencyRead 224
 - CurrencyReadAll 225
 - CurrencyUpsert 225
 - CurrentOrganizationRead 226
 - CustomRecordTypeRead 227
 - CustomRecordTypeReadAll 227
 - CustomRecordTypeUpsert 228
 - CustomWebLinkRead 229
 - CustomWebLinkReadAll 230
 - CustomWebLinkUpsert 231
 - CustomWebTabDelete 232
 - CustomWebTabRead 233
 - CustomWebTabReadAll 233
 - CustomWebTabUpsert 234
 - DetailPageCustomWebAppletRead 235
 - DetailPageCustomWebAppletReadAll 236
 - DetailPageCustomWebAppletUpsert 237
 - FieldManagementCreate 238
 - FieldManagementRead 239
 - FieldManagementReadAll 239
 - FieldManagementUpsert 240
 - HomepageCustomWebAppletRead 244
 - HomepageCustomWebAppletReadAll 245
 - HomepageCustomWebAppletUpsert 245
 - HomepageLayoutReadAll 242
 - HomeTabCustomWebAppletRead 247
 - HomeTabCustomWebAppletReadAll 247
 - HomeTabCustomWebAppletUpsert 248
 - HomepageLayoutUpsert 243
 - IndustryRead 249
 - IndustryReadAll 250
 - overview 210
 - PageLayoutFieldRead 251
 - PageLayoutFieldReadAll 252
 - PageLayoutFieldUpsert 253
 - PageLayoutRelatedInformationRead 254
 - PageLayoutRelatedInformationReadAll 254, 262
 - PageLayoutRelatedInformationUpsert 255
 - PageLayoutSectionReadAll 256
 - PageLayoutSectionUpsert 257
 - PicklistRead 258
 - PicklistReadAll 259
 - PicklistUpsert 260
 - RelatedInformationLayoutRead 261
 - RelatedInformationLayoutReadAll 262
 - RelatedInformationLayoutUpsert 263
 - RoleRead 264
 - RoleReadAll 265
 - RoleUpsert 265
 - SalesAssessmentTemplateRead 267
 - SalesAssessmentTemplateReadAll 268
 - SalesAssessmentTemplateUpsert 269
 - WorkflowActionSetRead 271
 - WorkflowActionSetReadForRuleSet 272
 - WorkflowActionSetUpsert 273
 - WorkflowRuleSetDelete 276
 - WorkflowRuleSetRead 277
 - WorkflowRuleSetReadAll 277
 - WorkflowRuleUpsert 278
- allocation object**
 - about 510
 - fields 510
 - list of values 511
 - methods 510
 - pick map fields 511
 - read-only fields 510
 - required fields 510
 - user key fields 511
- allotment usage object**
 - about 442
 - fields 443
 - methods 443
 - read-only fields 443
 - required fields 443
 - user keys 443
- API calls**
 - AccessProfileRead 214
 - AccessProfileReadAll 215
 - AccessProfileUpsert 215
 - ActionBarCustomWebAppletRead 216

- ActionBarCustomWebAppletReadAll 217
- ActionBarCustomWebAppletUpsert 218
- ActionBarLayoutReadAll 219
- ActionBarLayoutUpsert 220
- AuthAuthorizationUser method 140
- BulkOpImportCreateRequest 280
- BulkOpImportGetRequestDetail 284
- BulkOpImportSendData 283
- CascadingPicklistRead 221
- CascadingPicklistReadAll 222
- CascadingPicklistUpsert 223
- core methods 84
- CurrencyRead 224
- CurrencyReadAll 225
- CurrencyUpsert 225
- CurrentOrganizationRead 226
- CustomRecordTypeRead 227
- CustomRecordTypeReadAll 227
- CustomRecordTypeUpsert 228
- CustomWebLinkRead 229
- CustomWebLinkReadAll 230
- CustomWebLinkUpsert 231
- CustomWebTabDelete 232
- CustomWebTabRead 233
- CustomWebTabReadAll 233
- CustomWebTabUpsert 234
- Delete method (Web Services v1.0) 85
- Delete method (Web Services v2.0) 105
- DeleteChild method 88
- DeletedItemQueryPage method 142
- DeleteEvents method 149
- designing client applications 78
- DetailPageCustomWebAppletRead 235
- DetailPageCustomWebAppletReadAll 236
- DetailPageCustomWebAppletUpsert 237
- Echo Argument 85, 103
- Execute method 108
- ExportWebServiceCreateExportRequest 151
- ExportWebServiceMonitorExportRequest 154
- ExportWebServiceRetrieveExportAttachment 157
- FieldManagementCreate 238
- FieldManagementRead 239
- FieldManagementReadAll 239
- FieldManagementUpsert 240
- GetEvents method 159
- GetLists method 167
- GetMapping method 170
- GetPicklistValues method 173
- GetServerTime method 176
- HomepageCustomWebAppletRead 244
- HomepageCustomWebAppletReadAll 245
- HomepageCustomWebAppletUpsert 245
- HomepageLayoutReadAll 242
- HomepageLayoutUpsert 243
- HomeTabCustomWebAppletRead 247
- HomeTabCustomWebAppletReadAll 247
- HomeTabCustomWebAppletUpsert 248
- IndustryRead 249
- IndustryReadAll 250
- Insert method (Web Services v1.0) 89
- Insert method (Web Services v2.0) 112
- InsertChild method 90
- InsertOrUpdate method 90
- LoginHistoryQueryPage method 178
- LOVLanguageMode argument 104
- MergeRecords method 181
- MetadataChangeSummaryQueryPage method 184
- PageLayoutFieldRead 251
- PageLayoutFieldReadAll 252
- PageLayoutFieldUpsert 253
- PageLayoutRelatedInformationRead 254
- PageLayoutRelatedInformationReadAll 254, 262
- PageLayoutRelatedInformationUpsert 255
- PageLayoutSectionReadAll 256
- PageLayoutSectionUpsert 257
- PicklistRead 258
- PicklistReadAll 259
- PicklistUpsert 260
- QueryPage method (Web Services v1.0) 91
- QueryPage method (Web Services v2.0) 115
- RelatedInformationLayoutRead 261
- RelatedInformationLayoutReadAll 262
- RelatedInformationLayoutUpsert 263
- RoleRead 264
- RoleReadAll 265
- RoleUpsert 265
- SalesAssessmentTemplateRead 267
- SalesAssessmentTemplateReadAll 268
- SalesAssessmentTemplateUpsert 269
- SalesProcessQueryPage method 187
- Service API calls 139
- SetPassword method 193
- SetSessionTimeZone method 195
- Update method (Web Services v1.0) 101
- Update method (Web Services v2.0) 136
- UpdateCascadingPicklists method 201
- UpdateChild method 102
- UpdatePicklist method 197
- UserUsageQueryPage method 206
- ViewMode argument 104
- WorkflowActionSetRead 271
- WorkflowActionSetReadForRuleSet 272
- WorkflowActionSetUpsert 273
- WorkflowRuleSetDelete 276

- WorkflowRuleSetRead 277
- WorkflowRuleSetReadAll 277
- WorkflowRuleUpsert 278
- application object**
 - about 480
 - fields 481
 - list of values 482
 - methods 480
 - pick map fields 481
 - read-only fields 481
 - required fields 481
 - user key fields 481
- arguments**
 - Echo method 85, 103
 - LOVLanguageMode method 104
 - ViewMode method 104
- asset child object** 600
 - fields 601
 - pick map 601
 - read-only fields 601
 - required fields 601
 - user key fields 601
- asset object**
 - about 319, 443
 - fields 320, 444
 - filterable fields 321
 - list of values 321, 445
 - methods 320, 444
 - pick map fields 321, 444
 - read-only fields 320, 444
 - required fields 320, 444
 - status key 320
 - usage 319, 443
 - user key fields 321, 445
- attachment child object**
 - about 602
 - fields 602
 - read-only fields 602
 - required fields 602
 - user key fields 602
- attachment object** 426
- attachments** 639
- attendee call child object**
 - about 602
 - fields 603
 - pick map 603
 - read-only fields 603
 - required fields 603
 - user key fields 605
- audit fields** 288
- audit records, lifetime** 653
- audit trail object** 426
- AuthAuthorizationUser method**
 - about 140
 - arguments 141
 - call return value 141
 - sample SOAP request 141
- B**
- batch processing** 77
- best practices**
 - API calls 78
 - attachments 79
 - batch processing 77
 - error handling and logging 68
 - flushing of caches 65
 - indexed custom fields 80
 - integration design 65
 - integration management 81
 - integration performance 77
 - login history 79
 - moving customers between pods 81
 - opportunity product revenue records 68
 - outages 82
 - outages and failures 69
 - QueryPage method 79
 - service allotments 76
 - session management 77
 - setting and querying blank values 68
 - SSL certificates 82
 - stateless Web services 66
 - telephone numbers in Web services requests 74
 - user usage 79
 - Web services proxy classes 69
 - Web services requests 65
- blocked product object**
 - about 514
 - fields 514
 - methods 514
 - pick map fields 515
 - read-only fields 514
 - required fields 514
 - user key fields 515
- book child object**
 - about 605
 - fields 606
 - pick map 606
 - read-only fields 606
 - required fields 606
 - user key fields 606
- book object**
 - about 322, 445
 - child components 322
 - fields 323, 446
 - filterable fields 324
 - list of values 324, 446

- methods 322, 445
- pick map 324, 446
- read-only fields 323, 446
- required fields 323, 446
- status key 323
- user key fields 446
- book user object** 426
- BulkOpImportCreateRequest method**
 - arguments 282
 - call return value 283
- BulkOpImportGetRequestDetail method**
 - arguments 285
 - call return value 285
- BulkOpImportSendData method**
 - arguments 284
 - call return value 284
- business plan object**
 - about 512
 - fields 512
 - list of values 513
 - methods 512
 - pick map fields 513
 - read-only fields 512
 - required fields 512
 - user key fields 513

C

- campaign child object**
 - about 606
 - fields 607
 - pick map 607
 - picklists 607
 - read-only fields 607
 - required fields 607
 - user key fields 607
- campaign object**
 - about 325, 447
 - child components 325
 - fields 326, 447
 - filterable fields 332
 - list of values 333, 448
 - methods 325, 447
 - pick map field 329, 448
 - read-only fields 326, 447
 - required fields 326, 447
 - status key 328
 - user key fields 448
- CascadingPicklistRead method**
 - arguments 222
 - call return value 222
- CascadingPicklistReadAll method**
 - arguments 222
 - call return value 222

- CascadingPicklistUpsert method**
 - arguments 224
 - call return value 224
- category object**
 - about 448
 - methods 449
 - pick map field 449
 - read-only fields 449
 - required fields 449
 - user keys 449
- certification object**
 - about 482
 - fields 483
 - list of values 484
 - methods 482
 - pick map fields 483
 - read-only fields 483
 - required fields 483
 - user key fields 483
- certification request object**
 - about 484
 - fields 484
 - list of values 485
 - methods 484
 - pick map fields 485
 - read-only fields 484
 - required fields 484
 - user key fields 485
- child components**
 - account 291
 - activity 310
 - book 322
 - campaign 325
 - contact 334
 - current user 353
 - CustomObject1 355
 - household 408
 - lead 364
 - MedEd 405
 - opportunity 373
 - portfolio 414
 - service request 390
 - solution 395
 - user group 402
 - vehicle 421
- child objects**
 - account 590
 - account contact 591
 - account team 594
 - activity 596
 - address 598
 - asset 600
 - attachment 426, 602
 - attendee call 602

- audit trail 426
- book 605
- book user 426
- campaign 606
- competitor 426, 608
- contact 609
- contact role 610
- contact team 612
- custom object 613, 614
- delegated user 427, 615
- household team 427
- interests 428, 617
- invitee 428, 618
- lead 619
- login history 428
- multiple contact roles 428
- note 621
- opportunity 593, 621
- opportunity child 622
- opportunity team 428, 622
- portfolio team 429
- products detailed 429, 624
- promotional items dropped 429, 625
- quota 430
- recipient 430, 626
- related account 430
- related contact 430, 627
- revenue 430, 628
- sales assessment 629
- sales assessment value 630
- sample dropped 431, 631
- service request 632
- shared address 634
- sub book 431
- team 431
- user 637
- claim object**
 - about 558
 - fields 558
 - methods 558
 - pick map fields 559
 - picklists 559
 - read-only fields 558
 - user key fields 559
- client applications, designing**
 - API calls 78
 - batch processing 77
 - error handling 68
 - error logging 68
 - handling outages 82
 - Insert method 79
 - InsertorUpdate method 79
 - session management 77
 - session pooling 78
 - Update method 79
- client integrations**
 - examples 22
- client name identification** 41
- client-side session management** 57
- company-specific WSDL** 30
- competitor child object**
 - about 608
 - fields 608
 - pick map 608
 - picklists 609
 - read-only fields 608
 - required fields 608
 - user key fields 609
- competitor object**
 - about 426
 - fields 427
- contact best time object**
 - about 515
 - fields 516
 - list of values 517
 - methods 516
 - pick map fields 516
 - read-only fields 516
 - required fields 516
 - user key fields 517
- contact child object**
 - about 609
 - fields 609
 - pick map 610
 - read-only fields 609
 - required fields 609
 - user key fields 610
- contact object**
 - about 334, 450
 - child components 334
 - fields 335, 451
 - filterable fields 345
 - list of values 352, 452
 - methods 334, 451
 - objects not to use 353, 453
 - pick map 342, 452
 - read-only fields 335, 451
 - required fields 335, 451
 - status key 339
 - user key fields 452
- contact role child object**
 - about 610
 - fields 611
 - pick map 611
 - picklists 612
 - read-only fields 611
 - required fields 611
- contact state license object**

- about 517
- fields 518
- list of values 519
- methods 517
- pick map 518
- read-only fields 518
- required fields 518
- user key fields 518
- contact team child object**
 - about 612
 - fields 612
 - pick map 612
 - picklists 613
 - read-only fields 612
 - required fields 612
 - user key fields 613
- core methods**
 - Delete method (Web Services v1.0) 85
 - Delete method (Web Services v2.0) 105
 - DeleteChild method 88
 - Execute method 108
 - Insert method (Web Services v1.0) 89
 - Insert method (Web Services v2.0) 112
 - InsertChild method 90
 - InsertOrUpdate method 90
 - list of 84
 - MergeRecords 181
 - QueryPage method (Web Services v1.0) 91
 - QueryPage method (Web Services v2.0) 115
 - Update method (Web Services v1.0) 101
 - Update method (Web Services v2.0) 136
 - UpdateChild method 102
- core technologies** 21
 - about 21
- course enrollment object**
 - about 488
 - fields 489
 - list of values 490
 - methods 488
 - pick map fields 489
 - read-only fields 489
 - required fields 489
 - user key fields 490
- course object**
 - about 486
 - fields 486
 - list of values 488
 - methods 486
 - pick map fields 487
 - read-only fields 486
 - required fields 486
 - user key fields 487
- coverage object**
 - about 560
 - fields 561
 - methods 560
 - pick map fields 561
 - picklists 561
 - read-only fields 561
 - user key fields 561
- CRM On Demand**
 - custom fields 29
 - integration tag 30
- CRM On Demand Web Services** 22
 - communication with (figure) 22
 - filterable fields 290
 - parent and child relationships, about 29
 - reliability 25
 - Security 23
 - utilization 40
- currency and number fields** 34
- CurrencyRead method**
 - arguments 224
 - call return value 224
- CurrencyReadAll method**
 - arguments 225
 - call return value 225
- CurrencyUpsert method**
 - arguments 226
 - call return value 226
- current user object**
 - about 353
 - child components 353
 - fields 354
 - filterable fields 354
 - methods 353
 - read-only fields 354
 - required fields 354
- CurrentOrganizationRead method**
 - arguments 226
 - call return value 226
- custom fields** 29
- custom object child object**
 - about 613, 614
 - fields 613, 614
 - pick map 614
 - picklists 615
 - read-only fields 613, 614
 - required fields 613, 614
 - user key fields 613, 614
- customization-specific WSDL** 30
- CustomObject1 object**
 - about 355, 453
 - child components 355
 - fields 356, 454
 - filterable fields 362
 - list of values 364
 - methods 355, 454

- pick map 360, 455
- read-only fields 356, 454
- required fields 356, 454
- status key 358
- user keys 362, 456

CustomRecordTypeRead method

- arguments 227
- call return value 227

CustomRecordTypeReadAll method

- arguments 228
- call return value 228

CustomRecordTypeUpsert method

- arguments 229
- call return value 229

CustomWebLinkRead method

- arguments 230
- call return value 230

CustomWebLinkReadAll method

- arguments 230
- call return value 230

CustomWebLinkUpsert method

- arguments 232
- call return value 232

CustomWebTabDelete method

- arguments 232
- call return value 232

CustomWebTabRead method

- arguments 233
- call return value 233

CustomWebTabReadAll method

- arguments 234
- call return value 234

CustomWebTabUpsert method

- arguments 235
- call return value 235

D**damage object**

- about 562
- fields 562
- methods 562
- pick map fields 563
- read-only fields 562
- user key fields 563

Data Loader API calls

- available calls 280
- BulkOpImportCreateRequest 280
- BulkOpImportGetRequestDetail 284
- BulkOpImportSendData 283
- overview 280

date and time fields 33**deal registration object**

- about 490

- fields 491
- list of values 491
- methods 490
- pick map field 491
- read-only fields 491
- required fields 491
- user keys 491

dealer object

- about 418, 584
- fields 419, 585
- filterable fields 420
- list of values 586
- methods 418, 585
- pick map field 420, 585
- read-only fields 419, 585
- required fields 419, 585
- status key 420
- user keys 586

delegated user child object

- about 615
- fields 615
- read-only fields 615
- required fields 615
- user key fields 615

delegated user object 427**Delete method**

- arguments 87

Delete method (Web Services v1.0)

- usage 85

Delete method (Web Services v2.0)

- arguments 105
- sample SOAP request 106
- usage 105

DeleteChild method

- arguments 88
- usage 88

DeletedItemQueryPage method

- about 142
- arguments 145
- call return value 146
- sample SOAP request 146

DeleteEvents method

- about 149
- call return value 150
- sample SOAP request 150

DetailPageCustomWebAppletRead method

- arguments 236
- call return value 236

DetailPageCustomWebAppletReadAll method

- arguments 236

DetailPageCustomWebAppletUpsert method

- arguments 238
- call return value 238

E

- Echo argument** 85, 103
- email fields** 34
- Enable Web Services Access privilege** 49
- error handling** 68
- error logging** 68
- establishing a session** 53
- exam object**
 - about 492
 - fields 493
 - list of values 494
 - methods 492
 - pick map fields 493
 - read-only fields 493
 - required fields 493
 - user key fields 494
- exam registration object**
 - about 494
 - fields 495
 - list of values 496
 - methods 494
 - pick map fields 495
 - read-only fields 495
 - required fields 495
 - user key fields 496
- exchange rate history object**
 - about 616
 - fields 616
 - read-only fields 616
 - required fields 616
 - user key fields 616
- exchange rate object**
 - about 456
 - fields 457
 - methods 457
 - read-only fields 457
 - required fields 457
 - user key fields 457
- Execute method**
 - arguments 109
 - call return value 109
 - sample SOAP request 109
 - usage 108
- ExportWebServiceCreateExportRequest method**
 - arguments 151
 - call return value 152
 - sample SOAP request 152
- ExportWebServiceMonitorExportRequest method**
 - arguments 155
 - call return value 155
 - sample SOAP request 155

ExportWebServiceRetrieveExportAttachment method

- arguments 157
- call return value 157
- sample SOAP request 157

F

- features, what's new** 15

field types

- custom fields 29
- supported 30

FieldManagementCreate method

- arguments 238
- call return value 238

FieldManagementRead method

- arguments 239
- call return value 239

FieldManagementReadAll method

- arguments 240
- call return value 240

FieldManagementUpsert method

- arguments 242
- call return value 242

fields

- audit 288
- custom fields 29
- read-only, account child object 591
- read-only, account contact child object 592
- read-only, account object 292, 434
- read-only, account team child object 595
- read-only, accreditation object 477
- read-only, accreditation request object 479
- read-only, activity child object 596
- read-only, activity object 312, 437
- read-only, address child object 600
- read-only, allocation object 510
- read-only, allotment usage object 443
- read-only, application object 481
- read-only, asset child object 601
- read-only, asset object 320, 444
- read-only, attachment child object 602
- read-only, attendee call child object 603
- read-only, blocked product object 514
- read-only, book child object 606
- read-only, book object 323, 446
- read-only, business plan object 512
- read-only, campaign child object 607
- read-only, campaign object 326, 447
- read-only, category object 449
- read-only, certification object 483
- read-only, certification request object 484
- read-only, claim object 558
- read-only, competitor child object 608

- read-only, contact best time object 516
- read-only, contact child object 609
- read-only, contact object 335, 451
- read-only, contact role child object 611
- read-only, contact state license object 518
- read-only, contact team child object 612
- read-only, course enrollment object 489
- read-only, course object 486
- read-only, coverage object 561
- read-only, current user object 354
- read-only, custom object child object 613, 614
- read-only, CustomObject1 object 356, 454
- read-only, damage object 562
- read-only, deal registration object 491
- read-only, dealer object 419, 585
- read-only, delegated user child object 615
- read-only, exam object 493
- read-only, exam registration object 495
- read-only, exchange rate history object 616
- read-only, exchange rate object 457
- read-only, financial account holder object 565
- read-only, financial account holding object 567
- read-only, financial account object 564
- read-only, financial plan object 569
- read-only, financial product object 570
- read-only, financial transaction object 572
- read-only, fund object 497
- read-only, group object 458
- read-only, household object 410, 573
- read-only, insurance property object 576
- read-only, interests child object 617
- read-only, inventory audit report object 519
- read-only, inventory period object 522
- read-only, invitee child object 618
- read-only, involved party object 577
- read-only, lead child object 619
- read-only, lead object 365, 460
- read-only, MDF request object 498
- read-only, MedEd object 406, 523
- read-only, message response object 525
- read-only, messaging plan item object 529
- read-only, messaging plan item relation object 531
- read-only, messaging plan object 527
- read-only, modification tracking object 533
- read-only, note child object 621
- read-only, note object 372, 462
- read-only, objective object 534
- read-only, opportunity child object 593, 622
- read-only, opportunity object 374, 463
- read-only, opportunity team child object 623
- read-only, order item object 538
- read-only, order object 537
- read-only, partner object 500
- read-only, plan account object 539, 543
- read-only, plan contact object 541
- read-only, policy holder object 581
- read-only, policy object 579
- read-only, portfolio object 414, 583
- read-only, price list line item object 504
- read-only, price list object 502
- read-only, product category object 388
- read-only, product object 386, 465
- read-only, products detailed child object 624
- read-only, promotional items dropped child object 625
- read-only, recipient child object 626
- read-only, related contact child object 627
- read-only, revenue child object 628
- read-only, sales assessment child object 630
- read-only, sales assessment value child object 630
- read-only, sample disclaimer object 544
- read-only, sample dropped child object 631
- read-only, sample inventory object 546
- read-only, sample lot object 548
- read-only, sample request item object 551
- read-only, sample request object 550
- read-only, sample transaction object 552
- read-only, service allotment object 468
- read-only, service request child object 632
- read-only, service request object 391, 469
- read-only, shared address child object 636
- read-only, shared address object 441
- read-only, signature object 554
- read-only, solution object 395, 471
- read-only, SP request object 507
- read-only, special pricing product object 506
- read-only, territory object 398, 473
- read-only, transaction item object 556
- read-only, user child object 637
- read-only, user group object 403
- read-only, user object 401, 474
- read-only, vehicle object 422, 587
- required, account child object 591
- required, account contact child object 592
- required, account object 292, 434
- required, account team child object 595
- required, accreditation object 477
- required, accreditation request object 479
- required, activity child object 596
- required, activity object 312, 437
- required, address child object 600
- required, allocation object 510
- required, allotment usage object 443

- required, application object 481
- required, asset child object 601
- required, asset object 320, 444
- required, attachment child object 602
- required, attendee call child object 603
- required, blocked product object 514
- required, book child object 606
- required, book object 323, 446
- required, business plan object 512
- required, campaign child object 607
- required, campaign object 326, 447
- required, category object 449
- required, certification object 483
- required, certification request object 484
- required, competitor child object 608
- required, contact best time object 516
- required, contact child object 609
- required, contact object 335, 451
- required, contact role child object 611
- required, contact state license object 518
- required, contact team child object 612
- required, course enrollment object 489
- required, course object 486
- required, current user object 354
- required, custom object child object 613, 614
- required, CustomObject1 object 356, 454
- required, deal registration object 491
- required, dealer object 419, 585
- required, delegated user child object 615
- required, exam object 493
- required, exam registration object 495
- required, exchange rate history object 616
- required, exchange rate object 457
- required, financial account holder object 565
- required, financial account holding object 567
- required, financial account object 564
- required, financial plan object 569
- required, financial product object 570
- required, financial transaction object 572
- required, fund object 497
- required, group object 458
- required, household object 410, 573
- required, interests child object 617
- required, inventory audit report object 519
- required, inventory period object 522
- required, invitee child object 618
- required, lead child object 619
- required, lead object 365, 460
- required, MDF request object 498
- required, MedEd object 523
- required, message response object 525
- required, messaging plan item object 529
- required, messaging plan item relation object 531
- required, messaging plan object 527
- required, modification tracking object 533
- required, note child object 621
- required, note object 372, 462
- required, objective object 534
- required, opportunity child object 593, 622
- required, opportunity object 374, 463
- required, opportunity team child object 623
- required, order item object 538
- required, order object 537
- required, plan account object 539
- required, plan contact object 541
- required, plan opportunity object 543
- required, price list line item object 504
- required, price list object 502
- required, product category object 388
- required, product object 386, 465
- required, products detailed child object 624
- required, promotional items dropped child object 625
- required, recipient child object 626
- required, related contact child object 627
- required, revenue child object 628
- required, sales assessment child object 630
- required, sales assessment value child object 630
- required, sample disclaimer object 544
- required, sample dropped child object 631
- required, sample inventory object 546
- required, sample lot object 548
- required, sample request item object 551
- required, sample request object 550
- required, sample transaction object 552
- required, service allotment object 468
- required, service request child object 632
- required, service request object 391, 469
- required, shared address child object 636
- required, shared address object 441
- required, signature object 554
- required, solution object 395, 471
- required, SP request object 507
- required, special pricing product object 506
- required, territory object 398, 473
- required, transaction item object 556
- required, user child object 637
- required, user group object 403
- required, user object 401, 474
- required, vehicle object 422, 587
- filterable fields 290**
- financial account holder object**
 - about 565
 - fields 565

- methods 565
- pick map fields 566
- picklists 566
- read-only fields 565
- required fields 565
- user key fields 566

financial account holding object

- about 566
- fields 567
- methods 567
- pick map fields 567
- read-only fields 567
- required fields 567
- user key fields 568

financial account object

- about 563
- fields 564
- methods 563
- pick map fields 564
- picklists 564
- read-only fields 564
- required fields 564
- user key fields 564

financial plan object

- about 568
- fields 569
- methods 568
- pick map fields 569
- picklists 569
- read-only fields 569
- required fields 569
- user key fields 569

financial product object

- about 570
- fields 570
- methods 570
- pick map fields 570
- picklists 571
- read-only fields 570
- required fields 570
- user key fields 571

financial transaction object

- about 571
- fields 572
- methods 571
- pick map fields 572
- picklists 573
- read-only fields 572
- required fields 572
- user key fields 572

fund object

- about 496
- fields 497
- methods 496

- pick map field 497
- picklists 497
- read-only fields 497
- required fields 497
- user key fields 497

G

generating customized WSDL 30

GetEvents method

- about 159
- call return value 160
- sample SOAP requests 161

GetLists method

- about 167
- arguments 167
- call return value 167
- sample SOAP request 167

GetMapping method

- about 170
- call return value 171
- sample SOAP request 171

GetPickList method

- sample SOAP request 175

GetPicklistValues method

- about 173, 197, 201
- arguments 174
- call return value 174

GetServerTime method

- about 176
- call return value 177
- sample SOAP request 177

group object

- about 458
- methods 458
- pick map field 459
- read-only fields 458
- required fields 458
- user key fields 459

H

handling outages 82

HomepageCustomWebAppletRead method

- arguments 244
- call return value 245

HomepageCustomWebAppletReadAll method

- call return value 236

HomepageCustomWebAppletReadAll method

- arguments 245
- call return value 245

HomepageCustomWebAppletUpsert method

- arguments 246

- call return value 246
- HomepageLayoutReadAll method**
 - arguments 242
 - call return value 242
- HomepageLayoutUpsert method**
 - arguments 244
 - call return value 244
- HomeTabCustomWebAppletRead method**
 - arguments 247
 - call return value 247
- HomeTabCustomWebAppletReadAll method**
 - arguments 248
 - call return value 248
- HomeTabCustomWebAppletUpsert method**
 - arguments 249
 - call return value 249
- household object**
 - about 408, 573
 - child components 408
 - fields 410, 573
 - filterable fields 412
 - list of values 413, 421, 575
 - methods 409, 573
 - pick map field 412, 574
 - read-only fields 410, 573
 - required fields 410, 573
 - status key 411
 - user key fields 575
- household team object** 427
- I**
- Inbound Single Sign-On (SSO)** 64
- indexed custom fields** 80
- IndustryRead method**
 - arguments 250
 - call return value 250
- IndustryReadAll method**
 - arguments 251
 - call return value 251
- Insert method**
 - arguments 89
 - call return value 89
 - parent and child objects, effect on 89
- Insert method (Web Services v1.0)**
 - usage 89
- Insert method (Web Services v2.0)**
 - arguments 112
 - call return value 113
 - sample SOAP request 113
 - usage 112
- InsertChild method**
 - arguments 90
 - usage 90
- InsertOrUpdate method**
 - arguments 91
 - call return value 91
 - parent and child objects, effect on 91
 - usage 90
- insurance property object**
 - about 575
 - fields 576
 - methods 575
 - pick map fields 576
 - picklists 576
 - read-only fields 576
 - user key fields 576
- integration tag, viewing** 30
- integrations**
 - client integration examples 22
 - Web services session, integration requests 61
- interests child object** 428
 - about 617
 - fields 617
 - pick map 617
 - picklists 616, 618, 619
 - read-only fields 617
 - required fields 617
 - user key fields 617
- inventory audit report object**
 - about 519
 - fields 519
 - list of values 521
 - methods 519
 - pick map field 520
 - read-only fields 519
 - required fields 519
 - user key fields 520
- inventory period object**
 - about 521
 - fields 522
 - methods 521
 - pick map field 522
 - read-only fields 522
 - required fields 522
 - user key fields 522
- invitee child object** 428
 - about 618
 - fields 618
 - pick map 618
 - read-only fields 618
 - required fields 618
 - user key fields 619
- involved party object**
 - about 577
 - fields 577

methods 577
 pick map fields 578
 picklists 578
 read-only fields 577
 user key fields 578

isEncoded parameter 59

K

keys

status key, about 289
 status key, account object 296
 status key, activity object 314
 status key, asset object 320
 status key, book object 323
 status key, campaign object 328
 status key, contact object 339
 status key, CustomObject1 object 358
 status key, dealer object 420
 status key, household object 411
 status key, lead object 367
 status key, MedEd object 407
 status key, note object 372
 status key, opportunity object 377
 status key, portfolio object 415
 status key, product category object 389
 status key, product object 386
 status key, service request object 392
 status key, solution object 396
 status key, territory object 399
 status key, user group object 404
 status key, user object 401
 status key, vehicle object 423
 user key, about 288
 user key, allotment usage object 443
 user key, category object 449
 user key, price list line item object 505
 user key, price list object 503
 user key, product category object 389
 user key, product object 387, 466
 user key, sample disclaimer object 545
 user key, sample inventory object 547
 user key, sample lot object 549
 user key, sample transaction object 553
 user key, service allotment object 468
 user key, shared address object 442
 user key, signature object 555
 user key, territory object 473
 user key, transaction item object 557
 user key, user object 402, 475

L

lead child object

about 619

fields 619
 pick map 620
 read-only fields 619
 required fields 619
 user key fields 621

lead object

about 364, 459
 child components 364
 fields 365, 460
 methods 365, 459
 pick map fields 368, 460
 picklists 370, 461
 read-only fields 365, 460
 required fields 365, 460
 status key 367
 user key fields 461

lifetime of audit records 653

locale-dependent access 33

logging in

isEncoded parameter 59
 Single Sign-On 55, 60
 URL encoding 55, 59, 60
 Web services session 59

login history child object 428

LoginHistoryQueryPage method

about 178
 arguments 178
 call return value 179
 sample SOAP request 179

LOVLanguageMode argument 104

M

Manage Access to Web Services privilege 49

maximum number of records returned 654

maximum objects in a Web services request 654

maximum records returned 654

MDF request object

about 498
 fields 498
 list of values 499
 methods 498
 pick map fields 499
 read-only fields 498
 required fields 498
 user key fields 499

MedEd object

about 405, 522
 child components 405
 fields 406, 523
 filterable fields 407
 list of values 408, 524
 methods 406, 523

- pick map fields 407, 523
- read-only fields 406, 523
- required fields 523
- status key 407
- user key fields 407, 524
- MergeRecords method**
 - call return value 182
 - sample SOAP request 183
- message response object**
 - about 524
 - fields 525
 - list of values 526
 - methods 524
 - pick map fields 525
 - read-only fields 525
 - required fields 525
 - user key fields 526
- messaging plan item object**
 - about 528
 - fields 529
 - list of values 530
 - methods 529
 - pick map fields 529
 - read-only fields 529
 - required fields 529
 - user key fields 530
- messaging plan item relation object**
 - about 530
 - fields 531
 - list of values 532
 - methods 531
 - pick map fields 531
 - read-only fields 531
 - required fields 531
 - user key fields 532
- messaging plan object**
 - about 526
 - fields 527
 - list of values 528
 - methods 527
 - pick map fields 527
 - read-only fields 527
 - required fields 527
 - user key fields 528
- MetadataChangesSummaryQueryPage method**
 - sample SOAP request 185
- MetaDataChangeSummaryQueryPage method**
 - call return value 185
- MetadataChangeSummaryQueryPage method**
 - about 184
 - arguments 184
- methods called by**
 - account 292, 434
 - accreditation 477
 - accreditation request 479
 - activity 311, 437
 - allocation 510
 - allotment usage 443
 - application 480
 - asset 320, 444
 - blocked product 514
 - book 322, 445
 - business plan 512
 - campaign 325, 447
 - category 449
 - certification 482
 - certification request 484
 - claim 558
 - contact 334, 451
 - contact best time 516
 - contact state license 517
 - course 486
 - course enrollment 488
 - coverage 560
 - current user 353
 - CustomObject1 355, 454
 - damage 562
 - deal registration 490
 - dealer 418, 585
 - exam 492
 - exam registration 494
 - exchange rate 457
 - financial account 563
 - financial account holder 565
 - financial account holding 567
 - financial plan 568
 - financial product 570
 - financial transaction 571
 - fund 496
 - group 458
 - household 409, 573
 - insurance property 575
 - inventory audit report 519
 - inventory period 521
 - involved party 577
 - lead 365, 459
 - MDF request 498
 - MedEd 406, 523
 - message response 524
 - messaging plan 527
 - messaging plan item 529
 - messaging plan item relation 531
 - modification tracking 533
 - note 371, 461
 - objective 534

- opportunity 373, 463
 - order 536
 - order item 538
 - partner 500
 - plan account 539
 - plan contact 541
 - plan opportunity 542
 - policy 579
 - policy holder 581
 - portfolio 414, 582
 - price list 502
 - price list line item 504
 - product 386, 465
 - product category 388
 - sample disclaimer 544
 - sample inventory 546
 - sample lot 548
 - sample request 549
 - sample request item 551
 - sample transaction 552
 - service allotment 468
 - service request 390, 469
 - shared address 441
 - signature 554
 - solution 395, 471
 - SP request 507
 - special pricing product 505
 - territory 398, 472
 - transaction item 555
 - user 400, 474
 - user group 403
 - vehicle 422, 586
 - modification key**
 - about 289
 - modification tracking object**
 - about 532
 - fields 533
 - methods 533
 - read-only fields 533
 - required fields 533
 - user key fields 533
 - multiple contact roles child object** 428
 - multi-select picklists** 32
- N**
- NamedSearchSpec argument** 124
 - namespace, WSSE** 24
 - new features** 15
 - note child object**
 - about 621
 - fields 621
 - read-only fields 621
 - required fields 621
 - user key fields 621
 - note object**
 - about 371, 461
 - fields 372, 462
 - filterable fields 372
 - methods 371, 461
 - read-only fields 372, 462
 - required fields 372, 462
 - status key 372
 - user key fields 372, 462
 - number and currency fields** 34
- O**
- object display names** 649
 - objective object**
 - about 533
 - fields 534
 - list of values 536
 - methods 534
 - pick map fields 534
 - read-only fields 534
 - required fields 534
 - user key fields 535
 - objects**
 - filterable fields 290
 - modification key 289
 - parent and child relationships, about 29
 - pick maps 290
 - status keys 289
 - user keys 288
 - objects, exposed**
 - See parent objects; child objects
 - opportunity child object**
 - about 593, 621, 622
 - fields 593, 622
 - pick map 594
 - read-only fields 593, 622
 - required fields 593, 622
 - user key fields 594, 622
 - opportunity object**
 - about 372, 462
 - child components 373
 - fields 374, 463
 - list of value fields 385, 464
 - methods 373, 463
 - pick map field 378, 464
 - read-only fields 374, 463
 - required fields 374, 463
 - status key 377
 - user key fields 464
 - opportunity team child object** 428
 - about 622
 - fields 623

- pick map 623
 - picklists 624
 - read-only fields 623
 - required fields 623
 - user key fields 623
 - Oracle CRM On Demand Web Services**
 - Oracle CRM On Demand parent and child record types 25
 - pick maps 290
 - status keys 289
 - user keys 288
 - Oracle CRM On Demand Web Services Toolkit**
 - about 22
 - Oracle On Demand Web Services Toolkit**
 - CRM On Demand Web Services 22
 - order item object**
 - about 537
 - fields 538
 - list of values 539
 - methods 538
 - pick map fields 538
 - read-only fields 538
 - required fields 538
 - user key fields 539
 - order object**
 - about 536
 - fields 537
 - methods 536
 - pick map fields 537
 - read-only fields 537
 - required fields 537
 - user key fields 537
 - Outbound Single Sign-On (SSO)**
 - 63
 - outbound SSO** 59, 64
- P**
- PageLayoutFieldRead method**
 - arguments 251
 - call return value 252
 - PageLayoutFieldReadAll method**
 - arguments 252
 - call return value 252
 - PageLayoutFieldUpsert method**
 - arguments 253
 - call return value 254
 - PageLayoutRelatedInformationRead method**
 - arguments 254
 - call return value 254
 - PageLayoutRelatedInformationReadAll method**
 - arguments 255
 - call return value 255
 - PageLayoutRelatedInformationUpsert method**
 - arguments 256
 - call return value 256
 - PageLayoutSectionReadAll method**
 - arguments 257
 - call return value 257
 - PageLayoutSectionUpsert method**
 - arguments 258
 - call return value 258
 - parent and child objects**
 - Insert method, effect on 89
 - InsertOrUpdate method, effect on 91
 - relationships, about 29
 - Update method, effect of 101
 - parent and child record types**
 - about and list of 25
 - parent objects**
 - account 291, 433
 - accreditation 476
 - accreditation request 478
 - activity 310, 435
 - address 440
 - allocation 510
 - allotment usage 442
 - application 480
 - asset 319, 443
 - blocked product 514
 - book 322, 445
 - business plan 512
 - campaign 325, 447
 - category 448
 - certification 482
 - certification request 484
 - claim 558
 - contact 450
 - contact best time 515
 - contact object 334
 - contact state license 517
 - course 486
 - course enrollment 488
 - coverage 560
 - current user 353
 - damage 562
 - deal registration 490
 - dealer 418, 584
 - exam 492
 - exam registration 494
 - exchange rate 456
 - exchange rate history 616
 - financial account 563
 - financial account holder 565
 - financial account holding 566
 - financial plan 568

- financial product 570
- financial transaction 571
- fund 496
- group 458
- household 408, 573
- insurance property 575
- inventory audit report 519
- inventory period 521
- involved party 577
- lead 364, 459
- MDF request 498
- MedEd 405, 522
- message response 524
- messaging plan 526
- messaging plan item 528
- messaging plan item relation 530
- modification tracking 532
- note 371, 461
- objective 533
- opportunity 372, 462
- order 536
- order item 537
- partner 500
- plan account 539
- plan contact 540
- plan opportunity 542
- policy 578
- policy holder 580
- portfolio 413, 582
- price list 502
- price list line item 503
- product 385, 465, 466
- product category 388
- sample disclaimer 544
- sample inventory 545
- sample lot 547
- sample request 549
- sample request item 550
- sample transaction 552
- service request 390, 468
- signature 554
- solution 394, 470
- SP request 506
- special pricing product 505
- territory 398, 472
- transaction item 555
- user 399, 474
- user group 402
- vehicle 421
- parent objects (Web Services v1.0)**
 - list of 290
- partner object**
 - about 500
 - fields 500
 - list of values 501
 - methods 500
 - pick map fields 500
 - read-only fields 500
 - user key fields 501
- pick map**
 - account child object 591
 - account contact child object 592
 - account object 298, 434
 - account team child object 595
 - accreditation object 477
 - accreditation request object 479
 - activity child object 597
 - activity object 315, 438
 - address child object 600
 - allocation object 511
 - application object 481
 - asset child object 601
 - asset object 321, 444
 - attendee call child object 603
 - blocked product object 515
 - book child object 606
 - book object 324, 446
 - business plan object 513
 - campaign child object 607
 - campaign object 329, 448
 - category object 449
 - certification object 483
 - certification request object 485
 - claim object 559
 - competitor child object 608
 - contact best time object 516
 - contact child object 610
 - contact CustomObject1 360, 455
 - contact object 342, 452
 - contact role child object 611
 - contact state license object 518
 - contact team child object 612
 - course enrollment object 489
 - course object 487
 - coverage object 561
 - custom object child object 614
 - damage object 563
 - deal registration object 491
 - dealer object 420, 585
 - exam object 493
 - exam registration object 495
 - financial account holder object 566
 - financial account holding object 567
 - financial account object 564
 - financial plan object 569
 - financial product object 570
 - financial transaction object 572
 - fund object 497

- group object 459
- household object 412, 574
- insurance property object 576
- interests child object 617
- inventory audit report object 520
- inventory period object 522
- invitee child object 618
- involved party object 578
- lead child object 620
- lead object pick map fields 368, 460
- MDF request object 499
- MedEd object 407, 523
- message response object 525
- messaging plan item object 529
- messaging plan item relation object 531
- messaging plan object 527
- objective object 534
- opportunity child object 594
- opportunity object 378, 464
- opportunity team child object 623
- Oracle CRM On Demand pick maps,
 - about 290
- order item object 538
- order object 537
- partner object 500
- plan account object 540
- plan contact object 541
- plan opportunity object 543
- policy holder object 581
- policy object 579
- portfolio object 416, 583
- price list line item object 504
- price list object 503
- product category object 389
- product object 386, 465
- products detailed child object 624
- promotional items dropped child object 626
- related contact child object 627
- revenue child object 628
- sample disclaimer object 545
- sample dropped child object 631
- sample inventory object 546
- sample lot object 548
- sample request item object 551
- sample request object 550
- sample transaction object 553
- service request child object 633
- service request object 393, 469
- shared address child object 636
- shared address object 441
- signature object 554
- SP request object 507
- special pricing product object 506
- transaction item object 556
- user child object 637
- user group object 404
- user object 399, 401, 473, 475
- vehicle object 423, 587
- PicklistRead method**
 - arguments 259
 - call return value 259
- PicklistReadAll method**
 - arguments 259
 - call return value 260
- PicklistUpsert method**
 - arguments 261
 - call return value 261
- plan account object**
 - about 539
 - fields 539
 - methods 539
 - pick map fields 540
 - read-only fields 539, 543
 - required fields 539
 - user key fields 540
- plan contact object**
 - about 540
 - fields 541
 - methods 541
 - pick map fields 541
 - read-only fields 541
 - required fields 541
 - user key fields 542
- plan opportunity object**
 - about 542
 - fields 543
 - methods 542
 - pick map fields 543
 - required fields 543
 - user key fields 543
- policy holder object**
 - about 580
 - fields 581
 - methods 581
 - pick map fields 581
 - picklists 582
 - read-only fields 581
 - user key fields 582
- policy object**
 - about 578
 - fields 579
 - methods 579
 - pick map fields 579
 - picklists 580
 - read-only fields 579
 - user key fields 580
- portfolio object**
 - about 413, 582

- child components 414
 - fields 414, 583
 - filterable fields 416
 - list of values 417, 584
 - methods 414, 582
 - pick map fields 416, 583
 - read-only fields 414, 583
 - status key 415
 - user key fields 416, 583
 - portfolio team object**
 - about 429
 - price list line item object**
 - about 503
 - fields 504
 - methods 504
 - pick map field 504
 - picklists 505
 - read-only fields 504
 - required fields 504
 - user keys 505
 - price list object**
 - about 502
 - fields 502
 - methods 502
 - pick map field 503
 - picklists 503
 - read-only fields 502
 - required fields 502
 - user keys 503
 - privileges**
 - Enable Web Services Access 49
 - Manage Access to Web Services. 49
 - product category object**
 - about 388
 - fields 388, 449
 - methods 388
 - pick map field 389
 - read-only fields 388
 - required fields 388
 - status key 389
 - user keys 389
 - product object**
 - about 385, 465
 - fields 386, 465
 - methods 386, 465
 - pick map field 386, 465
 - picklists 387, 466
 - read-only fields 386, 465
 - required fields 386, 465
 - status key 386
 - user keys 387, 466
 - products detailed child object**
 - about 624
 - fields 624
 - pick map 624
 - read-only fields 624
 - required fields 624
 - user key fields 625
 - products detailed object 429**
 - promotional items dropped child object**
 - about 625
 - fields 625
 - pick map 626
 - read-only fields 625
 - required fields 625
 - user key fields 626
 - promotional items dropped object 429**
- Q**
- QueryPage method**
 - arguments 100
 - best practices 79
 - call return value 100
 - Oracle CRM On Demand query syntax,
 - about 94
 - querying data using Web Services 95
 - QueryPage method (Web Services v1.0)**
 - about 91
 - QueryPage method (Web Services v2.0)**
 - arguments 125
 - call return value 126
 - sample SOAP requests 126
 - usage 115
 - quota object 430**
- R**
- read-only fields**
 - account child object 591
 - account contact child object 592
 - account object 292, 434
 - account team child object 595
 - accreditation object 477
 - accreditation request object 479
 - activity child object 596
 - activity object 312, 437
 - address child object 600
 - allocation object 510
 - allotment usage object 443
 - application object 481
 - asset child object 601
 - asset object 320, 444
 - attachment child object 602
 - attendee call child object 603
 - blocked product object 514
 - book child object 606
 - book object 323, 446
 - business plan object 512

- campaign child object 607
- campaign object 326, 447
- category object 449
- certification object 483
- certification request object 484
- claim object 558
- competitor child object 608
- contact best time object 516
- contact child object 609
- contact object 335, 451
- contact role child object 611
- contact state license object 518
- contact team child object 612
- course enrollment object 489
- course object 486
- coverage object 561
- current user object 354
- custom object child object 613, 614
- CustomObject1 object 356, 454
- damage object 562
- deal registration object 491
- dealer object 419, 585
- delegated user child object 615
- exam object 493
- exam registration object 495
- exchange rate history object 616
- exchange rate object 457
- financial account holder object 565
- financial account holding object 567
- financial account object 564
- financial plan object 569
- financial product object 570
- financial transaction object 572
- fund object 497
- group object 458
- household object 410, 573
- insurance property object 576
- interests child object 617
- inventory audit report object 519
- inventory period object 522
- invitee child object 618
- involved party object 577
- lead child object 619
- lead object 365, 460
- MDF request object 498
- MedEd object 406, 523
- message response object 525
- messaging plan item object 529
- messaging plan item relation object 531
- messaging plan object 527
- modification tracking object 533
- note child object 621
- note object 372, 462
- objective object 534
- opportunity child object 593, 622
- opportunity object 374, 463
- opportunity team child object 623
- order item object 538
- order object 537
- partner object 500
- plan account object 539, 543
- plan contact object 541
- policy holder object 581
- policy object 579
- portfolio object 414, 583
- price list line item object 504
- price list object 502
- product category object 388
- product object 386, 465
- products detailed child object 624
- promotional items dropped child object 625
- recipient child object 626
- related contact child object 627
- revenue child object 628
- sales assessment child object 630
- sales assessment value child object 630
- sample disclaimer object 544
- sample dropped child object 631
- sample inventory object 546
- sample lot object 548
- sample request item object 551
- sample request object 550
- sample transaction object 552
- service allotment object 468
- service request child object 632
- service request object 391, 469
- shared address child object 636
- shared address object 441
- signature object 554
- solution object 395, 471
- SP request object 507
- special pricing product object 506
- territory object 398, 473
- transaction item object 556
- user child object 637
- user group object 403
- user object 401, 474
- vehicle object 422, 587
- recipient child object**
 - about 626
 - fields 626
 - read-only fields 626
 - required fields 626
 - user key fields 626
- recipient object** 430
- record types**
 - Oracle CRM On Demand parent and child record types 25

- related account object** 430
- related contact child object**
 - about 627
 - fields 627
 - pick map 627
 - picklists 628
 - read-only fields 627
 - required fields 627
 - user key fields 627
- related contact object** 430
- RelatedInformationLayoutRead method**
 - arguments 262
 - call return value 262
- RelatedInformationLayoutReadAll method**
 - arguments 263
 - call return value 263
- RelatedInformationLayoutUpsert method**
 - arguments 264
 - call return value 264
- release, what's new** 15
- reliability**
 - Web services reliability 25
- request rate limit** 654
- request size limit** 654
- required fields**
 - account child object 591
 - account contact child object 592
 - account object 292, 434
 - account team child object 595
 - accreditation object 477
 - accreditation request object 479
 - activity child object 596
 - activity object 312, 437
 - address child object 600
 - allocation object 510
 - allotment usage object 443
 - application object 481
 - asset child object 601
 - asset object 320, 444
 - attachment child object 602
 - attendee call child object 603
 - blocked product object 514
 - book child object 606
 - book object 323, 446
 - business plan object 512
 - campaign child object 607
 - campaign object 326, 447
 - category object 449
 - certification object 483
 - certification request object 484
 - competitor child object 608
 - contact best time object 516
 - contact child object 609
 - contact object 335, 451
 - contact role child object 611
 - contact state license object 518
 - contact team child object 612
 - course enrollment object 489
 - course object 486
 - current user object 354
 - custom object child object 613, 614
 - CustomObject1 object 356, 454
 - deal registration object 491
 - dealer object 419, 585
 - delegated user child object 615
 - exam object 493
 - exam registration object 495
 - exchange rate history object 616
 - exchange rate object 457
 - financial account holder object 565
 - financial account holding object 567
 - financial account object 564
 - financial plan object 569
 - financial product object 570
 - financial transaction object 572
 - fund object 497
 - group object 458
 - household object 410, 573
 - interests child object 617
 - inventory audit report object 519
 - inventory period object 522
 - invitee child object 618
 - lead child object 619
 - lead object 365, 460
 - MDF request object 498
 - MedEd object 523
 - message response object 525
 - messaging plan item object 529
 - messaging plan item relation object 531
 - messaging plan object 527
 - modification tracking object 533
 - note child object 621
 - note object 372, 462
 - objective object 534
 - opportunity child object 593, 622
 - opportunity object 374, 463
 - opportunity team child object 623
 - order item object 538
 - order object 537
 - plan account object 539
 - plan contact object 541
 - plan opportunity object 543
 - price list line item object 504
 - price list object 502
 - product category object 388
 - product object 386, 465
 - products detailed child object 624
 - promotional items dropped child object 625

- recipient child object 626
- related contact child object 627
- revenue child object 628
- sales assessment child object 630
- sales assessment value child object 630
- sample disclaimer object 544
- sample dropped child object 631
- sample inventory object 546
- sample lot object 548
- sample request item object 551
- sample request object 550
- sample transaction object 552
- service allotment object 468
- service request child object 632
- service request object 391, 469
- shared address child object 636
- shared address object 441
- signature object 554
- solution object 395, 471
- SP request object 507
- special pricing product object 506
- territory object 398, 473
- transaction item object 556
- user child object 637
- user group object 403
- user object 401, 474
- vehicle object 422, 587
- revenue child object**
 - about 628
 - fields 628
 - pick map 628
 - picklists 629
 - read-only fields 628
 - required fields 628
 - user key fields 629
- revenue object** 430
- RoleRead method**
 - arguments 265
 - call return value 265
- RoleReadAll method**
 - arguments 265
 - call return value 265
- RoleUpsert method**
 - arguments 267
 - call return value 267
- S**
- sales assessment child object** 629
 - fields 630
 - read-only fields 630
 - required fields 630
 - user key fields 630
- sales assessment value child object** 630
 - fields 630
 - read-only fields 630
 - required fields 630
 - user key fields 631
- SalesAssessmentTemplateRead method**
 - arguments 268
 - call return value 268
- SalesAssessmentTemplateReadAll method**
 - arguments 269
 - call return value 269
- SalesAssessmentTemplateUpsert method**
 - arguments 271
 - call return value 271
- SalesProcessQueryPage method**
 - about 187
 - arguments 187
 - call return value 188
 - sample SOAP requests 189
- sample disclaimer object**
 - about 544
 - fields 544
 - methods 544
 - pick map field 545
 - picklists 545
 - read-only fields 544
 - required fields 544
 - user keys 545
- sample dropped child object**
 - about 631
 - fields 631
 - pick map 631
 - read-only fields 631
 - required fields 631
 - user key fields 632
- sample dropped object** 431
- sample inventory object**
 - about 545
 - fields 546
 - methods 546
 - pick map field 546
 - read-only fields 546
 - required fields 546
 - user keys 547
- sample lot object**
 - about 547
 - fields 548
 - methods 548
 - pick map field 548
 - read-only fields 548
 - required fields 548
 - user keys 549
- sample request item object**
 - about 550
 - fields 551

- list of values 551
- methods 551
- pick map fields 551
- read-only fields 551
- required fields 551
- user key fields 551
- sample request object**
 - about 549
 - fields 550
 - methods 549
 - pick map fields 550
 - read-only fields 550
 - required fields 550
 - user key fields 550
- sample transaction object**
 - about 552
 - fields 552
 - methods 552
 - pick map field 553
 - picklists 553
 - read-only fields 552
 - required fields 552
 - user keys 553
- security**
 - Web services security 23
- server-side session management** 54
- service allotment object**
 - about 466
 - fields 468
 - methods 468
 - read-only fields 468
 - required fields 468
 - user keys 468
- service allotments**
 - best practices 76
 - overview 45
 - Web services concurrent request allotment 45
 - Web services operations allotment 45
- Service API calls** 139
 - AuthAuthorizationUser 140
 - DeletedItemQueryPage 142
 - DeleteEvents 149
 - ExportWebServiceCreateExportRequest 151
 - ExportWebServiceMonitorExportRequest 154
 - ExportWebServiceRetrieveExportAttachment 157
 - GetEvents 159
 - GetLists 167
 - GetMapping 170
 - GetPicklistValues 173
 - GetServerTime 176
 - LoginHistoryQueryPage 178
 - MetadataChangeSummaryQueryPage 184
 - SalesProcessQueryPage 187
 - SetPassword 193
 - SetSessionTimeZone 195
 - UpdateCascadingPicklists 201
 - UpdatePicklist 197
 - UserUsageQueryPage 206
- service request child object**
 - about 632
 - fields 632
 - pick map 633
 - read-only fields 632
 - required fields 632
 - user key fields 634
- service request object**
 - about 390, 468
 - child components 390
 - fields 391, 469
 - filterable fields 393
 - methods 390, 469
 - pick map 393, 469
 - picklists 394, 470
 - read-only fields 391, 469
 - required fields 391, 469
 - status key 392
 - user key fields 470
- session limit quota** 653
- session management** 53, 77
- session management, client-side** 57
- session management, server-side** 54
- session pooling** 78
- session time-outs** 77
- SetPassword** 193
- SetPassword method**
 - sample SOAP request 194
- SetSessionTimeZone method**
 - about 195
 - call return value 196
 - sample SOAP request 196
- setting up, Web services** 49
- setting up, Web services session**
 - establishing and managing 53
 - integration requests 61
 - limits 653
 - logging in 59
 - logging off 61
- shared address child object** 634
 - fields 636
 - pick map 636
 - picklists 636
 - read-only fields 636
 - required fields 636
 - user key fields 636
- shared address object**

- about 440
 - fields 441
 - list of values 442
 - methods 441
 - pick map fields 441
 - read-only fields 441
 - required fields 441
 - user keys 442
 - signature object**
 - about 554
 - fields 554
 - methods 554
 - pick map field 554
 - read-only fields 554
 - required fields 554
 - user keys 555
 - Single Sign-On (SSO)**
 - 63
 - logging in using 55, 60
 - outbound SSO 59, 64
 - solution object**
 - about 394, 470
 - child components 395
 - fields 395, 471
 - filterable fields 397
 - list of value fields 397, 472
 - methods 395, 471
 - read-only fields 395, 471
 - required fields 395, 471
 - status key 396
 - user key fields 471
 - SP request object**
 - about 506
 - fields 507
 - list of values 508
 - methods 507
 - pick map fields 507
 - read-only fields 507
 - required fields 507
 - user key fields 508
 - special pricing product object**
 - about 505
 - fields 506
 - methods 505
 - pick map fields 506
 - read-only fields 506
 - required fields 506
 - user key fields 506
 - stateful Web services** 50, 58, 77
 - stateless Web services** 50, 54, 66
 - status key**
 - about 289
 - account object 296
 - activity object 314
 - asset object 320
 - book object 323
 - campaign object 328
 - contact object 339
 - CustomObject1 object 358
 - dealer object 420
 - household object 411
 - lead object 367
 - MedEd object 407
 - note object 372
 - opportunity object 377
 - portfolio object 415
 - product category object 389
 - product object 386
 - service request object 392
 - solution object 396
 - territory object 399
 - user group object 404
 - user object 401
 - vehicle object 423
 - sub book object** 431
- T**
- team object** 431
 - territory object**
 - about 398, 472
 - fields 398, 473
 - methods 398, 472
 - picklists 399, 473
 - read-only fields 398, 473
 - required fields 398, 473
 - status key 399
 - user key 473
 - time and date fields** 33
 - transaction item object**
 - about 555
 - fields 556
 - methods 555
 - pick map field 556
 - read-only fields 556
 - required fields 556
 - user keys 557
- U**
- Update method**
 - arguments 101
 - call return value 102
 - parent and child objects, effect of 101
 - Update method (Web Services v1.0)**
 - usage 101
 - Update method (Web Services v2.0)**
 - arguments 136
 - call return value 137

- sample SOAP request 137
- usage 136
- UpdateCascadingPicklists method**
 - arguments 202
 - call return value 203
 - sample SOAP request 203
 - sample SOAP requests 203
- UpdateChild method**
 - arguments 102
 - usage 102
- UpdatePicklist method**
 - arguments 198
 - call return value 198
 - sample SOAP requests 198
- URL parameterization** 81
- user child object**
 - about 637
 - fields 637
 - pick map 637
 - read-only fields 637
 - required fields 637
 - user key fields 637
- user group object**
 - about 402
 - child components 402
 - fields 403, 422, 458, 587
 - filterable fields 404
 - methods 403
 - pick map field 404
 - read-only fields 403
 - required fields 403
 - status key 404
- user key**
 - inventory period object 522
 - MedEd object 524
 - message response object 526
 - messaging plan item object 530
 - messaging plan item relation object 532
 - messaging plan object 528
 - modification tracking object 533
 - objective object 535
 - order item object 539
 - order object 537
 - plan account object 540
 - plan contact object 542
 - plan opportunity object 543
 - sample request item object 551
 - sample request object 550
- user keys**
 - about 288
 - account object 435
 - accreditation object 478
 - accreditation request object 480
 - activity object 439
 - allocation object 511
 - allotment usage object 443
 - application object 481
 - asset object 445
 - blocked product object 515
 - book object 446
 - business plan object 513
 - campaign object 448
 - category object 449
 - certification object 483
 - certification request object 485
 - claim object 559
 - contact best time object 517
 - contact object 452
 - contact state license object 518
 - course enrollment object 490
 - course object 487
 - coverage object 561
 - CustomObejct1 object 456
 - damage object 563
 - deal registration object 491
 - dealer object 586
 - exam object 494
 - exam registration object 496
 - exchange rate history object 616
 - exchange rate object 457
 - financial account holder object 566
 - financial account holding object 568
 - financial account object 564
 - financial plan object 569
 - financial product object 571
 - financial transaction object 572
 - fund object 497
 - group object 459
 - household object 575
 - insurance property object 576
 - inventory audit report object 520
 - involved party object 578
 - lead object 461
 - MDF request object 499
 - note object 462
 - opportunity object 464
 - partner object 501
 - policy holder object 582
 - policy object 580
 - portfolio object 583
 - price list line item object 505
 - price list object 503
 - product category object 389
 - product object 387, 466
 - sample disclaimer object 545
 - sample inventory object 547
 - sample lot object 549
 - sample transaction object 553

- service allotment object 468
- service request object 470
- shared address object 442
- signature object 555
- solution object 471
- SP request object 508
- special pricing product object 506
- territory object 473
- transaction item object 557
- user object 402, 475
- vehicle object 587
- user object**
 - about 399, 474
 - fields 400, 474
 - methods 400, 474
 - pick map field 399, 401, 473, 475
 - read-only fields 401, 474
 - required fields 401, 474
 - status key 401
 - user keys 402, 475
- UserNameToken profile** 24
- UserUsageQueryPage method**
 - about 206
 - arguments 207
 - call return value 207
 - sample SOAP request 208
- utilization**
 - Web services reliability 40
- V**
- vehicle object**
 - about 421
 - child components 421
 - filterable fields 423
 - methods 422, 586
 - pick map field 423, 587
 - picklists 424, 588
 - read-only fields 422, 587
 - required fields 422, 587
 - status key 423
 - user key fields 587
- ViewMode argument** 104
- W**
- Web service request rate** 653
- Web service request size** 654
- Web service session inactivity time limit** 654
- Web services**
 - core technologies 21
 - custom fields 29
 - integration tag 30
 - reliability 25
 - security 23
 - utilization 40
- Web services privilege**
 - Enable Web Services Access 49
 - Manage Access to Web Services 49
- Web services proxy classes** 69
- Web Services R16 Compatibility Mode setting** 50
- Web services session**
 - establishing and managing 53
 - integration requests 61
 - logging in 59
 - logging off 61
 - maximum objects in Web Services request 654
 - maximum records returned 654
 - request rate limit 654
 - request size limit 654
- Web Services Toolkit**
 - See Oracle CRM On Demand Web Services Toolkit
- WorkflowActionSetRead method**
 - arguments 271
 - call return value 272
- WorkflowActionSetReadReadForRuleSet method**
 - arguments 273
 - call return value 273
- WorkflowActionSetUpsert method**
 - arguments 276
 - call return value 276
- WorkflowRuleDelete method**
 - arguments 276
- WorkflowRuleSetDelete method**
 - call return value 277
- WorkflowRuleSetRead method**
 - arguments 277
 - call return value 277
- WorkflowRuleSetReadAll method**
 - arguments 278
 - call return value 278
- WorkflowRuleSetUpsert method**
 - arguments 280
 - call return value 280
- WSDL, generating company-specific** 30
- WS-I Basic Security Profile** 23, 24
- WSSE namespace** 24, 50

