

# **Oracle Web Services On Demand Guide**

Version 24.0 (Oracle CRM On Demand Release 36)  
March 2018

---

Copyright © 2005, 2018 Oracle. 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 END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

#### Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

#### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

# Contents

---

## Chapter 1: What's New in This Release

## Chapter 2: Overview of Web Services On Demand

About Web Services	17
Web Services Core Technologies	17
Oracle CRM On Demand Web Services Toolkit	18
Oracle CRM On Demand Web Services and Integration with Oracle CRM On Demand	18
Web Services Security	19
Web Services Reliability	21
Web Services and the Oracle CRM On Demand Objects	21
Web Service APIs	21
About Parent-Child Relationships	25
Web Services On Demand and Custom Fields	25
Field Types Supported by Oracle CRM On Demand	26
Web Services v1.0	26
Web Services v2.0	27
Special Search Fields	28
Support for Multi-Select Picklists	29
Locale-Dependent Access to Oracle CRM On Demand	29
Validation of Email Fields	30
Mapping Primary Address Fields Using Web Services	32
Support for Web Link Fields	36
Support for Concatenated Fields	39
Support for Maskable Fields	39
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	42
Sending the Client Name in Stateful Web Services	43
Web Services R16 Compatibility Mode	44
About Service Allotments	45
Determining Current Usage	46

- Determining Historical Allotment Usage 46
- When a Service Allotment Is Reached 47
- Calculation of Allotment Usage 47

## **Chapter 3: Getting Started with Web Services**

- Accessing Data in Oracle CRM On Demand 49
- Requesting Web Services Integration 49
  - Web Services Privileges 49
  - Enabling Stateless Web Services Support 50
- Downloading WSDL Files and Schema Files 50
  - Downloading WSDL Files 51
  - Downloading Schema Files 52
- Incorporating WSDL Files into the Development Environment 53

## **Chapter 4: Establishing and Managing the Web Services Session**

- About Establishing and Managing the Web Services Session 55
  - Oracle CRM On Demand Stateful and Stateless Request Support 55
- Using Stateless Web Service Requests 56
  - Enabling Support for Stateless Requests 56
  - Oracle CRM On Demand Stateless Authentication Mechanisms 56
  - Stateless Server Affinity Support 58
  - Maintenance of a Stateless Java Session 58
- Using Stateful Web Services Requests 59
  - Oracle CRM On Demand Stateful Authentication Mechanisms 60
  - Logout of the Web Services Session 62
- Integration Requests for the Web Services Session 63
- Comparison of Stateless Versus Stateful Web Services Transactions 64
- Single Sign-On 65
  - Outbound SSO 65
  - SSO Token Validation 65
  - Inbound SSO 65

## **Chapter 5: Best Practices for Designing Client Application**

- Best Practices for Integration Design 67
  - Sending of Web Services Requests to Oracle CRM On Demand 67
  - Flushing of Caches 67
  - Using Stateless Web Services 68

Setting and Querying Blank Values	70
Working with Opportunity Product Revenue Records	70
Error Handling and Logging	70
Handling Outages and Failures	71
Understanding Web Services Transactions	72
Best Practices for Generating Web Services Proxy Classes in Java Environments	72
Avoiding Proxy Class Compilation Failure Due to the Java 64KB Limit	73
Entering Telephone Number Values through Web Services	76
Best Practices for Adhering to Web Service Allotments	79
Working with Record Ownership Modes	80
Considerations when Rejecting Leads	80
Best Practices for Integration Performance	80
Batch Processing	80
Session Management and Pooling	81
API Calls	82
Handling Bad Events in Integration Event Queues	85
Best Practices for Integration Management	86
Moving Customers Between Pods	86
Handling Outages	86
Maintaining SSL Certificates	86

## Chapter 6: Web Services On Demand API Calls

Web Services On Demand API Calls	89
Considerations When Deleting Child Records	89
Web Services v1.0 API Calls	90
User Keys	91
The Echo Argument	91
Delete	91
DeleteChild	92
Insert	93
InsertChild	94
InsertOrUpdate	95
QueryPage	96
Update	106
UpdateChild	107
Web Services v2.0 API Calls	107
The Echo Argument	108
The ViewMode Argument	109
The LOVLanguageMode Argument	109
The Field-level Replace Operation	110

Delete	112
Execute	116
Insert	120
QueryPage	123
Update	145
Service API Calls	147
AuditTrailQueryPage	149
AuthAuthorizationUser	155
DeletedItemQueryPage	157
DeleteEvents	163
ExportWebServiceCreateExportRequest	166
ExportWebServiceMonitorExportRequest	169
ExportWebServiceRetrieveExportAttachment	171
GetEvents	173
GetLists	182
GetMapping	185
GetPicklistValues	189
GetServerTime	192
LeadConvert	194
LoginHistoryQueryPage	200
MergeRecords	203
MetadataChangeSummaryQueryPage	207
SalesProcessQueryPage	210
SetPassword	216
SetSessionTimeZone	218
UpdatePicklist	220
UpdateCascadingPicklists	224
UserUsageQueryPage	229
Administrative Services API Calls	233
Access to the Administrative Services APIs for Nonadministrative Users	238
Record Ownership Mode	239
AccessProfileRead	241
AccessProfileReadAll	242
AccessProfileUpsert	243
ActionBarCustomWebAppletRead	244
ActionBarCustomWebAppletReadAll	245
ActionBarCustomWebAppletUpsert	246
ActionBarLayoutReadAll	247
ActionBarLayoutUpsert	248
AssignmentRuleRead	250
AssignmentRuleReadAll	251
AssignmentRuleUpsert	251

AssignmentRuleGroupRead	253
AssignmentRuleGroupReadAll	254
AssignmentRuleGroupUpsert	254
CascadingPicklistRead	256
CascadingPicklistReadAll	257
CascadingPicklistUpsert	257
ConcatenatedFieldRead	258
ConcatenatedFieldReadAll	259
ConcatenatedFieldUpsert	260
CurrencyRead	261
CurrencyReadAll	262
CurrencyUpsert	262
CurrentOrganizationRead	264
CustomHTMLHeadTagRead	269
CustomHTMLHeadTagUpsert	270
CustomRecordTypeRead	270
CustomRecordTypeReadAll	271
CustomRecordTypeUpsert	272
CustomWebLinkRead	274
CustomWebLinkReadAll	275
CustomWebLinkUpsert	275
CustomWebTabDelete	277
CustomWebTabRead	278
CustomWebTabReadAll	278
CustomWebTabUpsert	279
DetailPageCustomWebAppletRead	280
DetailPageCustomWebAppletReadAll	281
DetailPageCustomWebAppletUpsert	282
DynamicLayoutRead	283
DynamicLayoutReadAll	284
DynamicLayoutUpsert	285
EnableModificationTrackingRead	286
EnableModificationTrackingReadAll	287
EnableModificationTrackingUpsert	287
FieldAuditSetupRead	288
FieldAuditSetupReadAll	289
FieldAuditSetupUpsert	289
FieldManagementCreate	290
FieldManagementRead	291
FieldManagementReadAll	292
FieldManagementUpsert	293
HomepageLayoutReadAll	295

HomepageLayoutUpsert	296
HomepageCustomWebAppletRead	297
HomepageCustomWebAppletReadAll	298
HomepageCustomWebAppletUpsert	299
HomeTabCustomWebAppletRead	301
HomeTabCustomWebAppletReadAll	301
HomeTabCustomWebAppletUpsert	302
IndustryRead	304
IndustryReadAll	304
IntegrationEventQueueRead	305
IntegrationEventQueueReadAll	306
IntegrationEventQueueUpsert	307
ListRead	309
ListReadAll	312
ListAccessAndOrderRead	312
ListAccessAndOrderReadAll	313
ListAccessAndOrderUpsert	314
PageLayoutFieldRead	315
PageLayoutFieldReadAll	315
PageLayoutFieldUpsert	316
PageLayoutRelatedInformationRead	317
PageLayoutRelatedInformationReadAll	318
PageLayoutRelatedInformationUpsert	319
PageLayoutSectionReadAll	320
PageLayoutSectionUpsert	321
PicklistRead	322
PicklistReadAll	323
PicklistUpsert	324
PicklistValueGroupRead	325
PicklistValueGroupReadAll	326
PicklistValueGroupUpsert	327
ProcessAdministrationRead	328
ProcessAdministrationReadAll	328
ProcessAdministrationUpsert	329
RelatedInformationLayoutRead	331
RelatedInformationLayoutReadAll	332
RelatedInformationLayoutUpsert	333
ReportFolderRead	334
ReportFolderReadAll	335
ReportFolderUpsert	336
RoleRead	338
RoleReadAll	339



RoleUpsert	339
SalesAssessmentTemplateRead	341
SalesAssessmentTemplateReadAll	342
SalesAssessmentTemplateUpsert	343
SalesCategoryRead	345
SalesCategoryReadAll	345
SalesCategoryUpsert	346
SearchLayoutRead	347
SearchLayoutReadAll	348
SearchLayoutUpsert	348
SSOTokenRead	351
WorkflowActionSetRead	352
WorkflowActionSetReadForRuleSet	353
WorkflowActionSetUpsert	355
WorkflowActionSetInsertOrUpdate	359
WorkflowRuleSetDelete	360
WorkflowRuleSetRead	361
WorkflowRuleSetReadAll	362
WorkflowRuleSetUpsert	363
WorkflowRuleSetInsertOrUpdate	365
Data Loader API Calls	366
BulkOpImportCreateRequest	366
BulkOpImportSendData	369
BulkOpImportGetRequestDetail	370

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

Reference Information About the Parent Objects	373
Field Names in WSDL Files	374
Required and Read-Only Fields	374
Oracle CRM On Demand User Keys	374
Audit Fields	375
Oracle CRM On Demand Status Keys	375
Oracle CRM On Demand Pick Maps	376
Filterable Fields	377
Core Parent Objects (Web Services v1.0)	377
Account	378
Activity	397
Asset	406
Book	409
Campaign	412

Contact	421
Current User	440
CustomObject1 - CustomObject3	442
Lead	451
Note	458
Opportunity	459
Product	472
Product Category	475
Service Request	477
Solution	481
Territory	484
User	486
User Group	489
Life Sciences Edition Parent Objects (Web Services v1.0)	491
Event	492
Financial Services Edition Parent Objects (Web Services v1.0)	495
Household	495
Portfolio	499
Automotive Edition Parent Objects (Web Services v1.0)	504
Dealer	504
Vehicle	507
Child Objects (Web Services v1.0)	510
Address	511
Attachment	512
Audit Trail	512
BookUser	512
Competitor	512
DelegatedUser	513
HouseholdTeam	513
Interests	514
Invitee	514
Login History	514
Multiple Contact Roles	514
OpportunityTeam	514
PortfolioTeam	515
ProductsDetailed	515
PromotionalItemsDropped	515
Quota	516
Recipient	516
Related Account	516
Related Contact	516

Revenue	516
SampleDropped	517
Shared Address	517
SubBook	517
Team	518
Core Parent Objects (Web Services v2.0)	518
Account	519
Activity	522
Address	527
Address Location	539
Allotment Usage	540
Asset	541
Book	543
Campaign	544
Category	546
Contact	547
CustomObject	551
Division	555
Exchange Rate	557
Group	558
Lead	559
Modification Tracking	562
Note	563
Opportunity	564
Period	566
Product	567
Service Allotment	569
Service Request	571
Social Media	573
Social Profile	574
Solution	576
Territory	577
User	579
Partner Relationship Management Edition Objects (Web Services v2.0)	581
Accreditation	582
Accreditation Request	584
Application	585
Certification	587
Certification Request	589
Course	591
Course Enrollment	593
Deal Registration	595

Exam	597
Exam Registration	599
Fund	601
MDF Request	603
Partner	605
Price List	607
Price List Line Item	608
Special Pricing Product	610
Special Pricing Request	611
Life Sciences Edition Parent Objects (Web Services v2.0)	613
Allocation	615
Blocked Product	616
Business Plan	618
Call Activity History	620
Call Product Detail	621
Contact Best Times	623
Contact State License	624
Event	626
HCP Contact Allocation	628
Inventory Audit Report	630
Inventory Period	632
Message Response	633
Messaging Plan	635
Messaging Plan Item	637
Messaging Plan Item Relation	639
Objective	641
Order	643
Order Item	645
Plan Account	646
Plan Contact	648
Plan Opportunity	650
Product Indication	651
Related Disclaimer	652
Sample Disclaimer	654
Sample Inventory	657
Sample Lot	659
Sample Request	661
Sample Request Item	663
Sample Transaction	664
Signature	667
Transaction Item	668
Financial Services Edition Parent Objects (Web Services v2.0)	670

Claim	671
Coverage	673
Damage	675
Financial Account	676
Financial Account Holder	678
Financial Account Holding	680
Financial Plan	681
Financial Product	683
Financial Transaction	684
Household	686
Insurance Property	688
Involved Party	690
Policy	691
Policy Holder	693
Portfolio	695
Automotive Edition Parent Objects (Web Services v2.0)	697
Dealer	697
Vehicle	699
Child Objects (Web Services v2.0)	701
Account	703
Account Contact	704
Account Opportunity	706
Activity	707
Address	709
Asset	712
Attachment	714
Attendee Call	715
Book	717
Book User	718
Campaign	720
Competitor	721
Contact	722
Contact Role	724
Custom Object 1 through 3	725
Custom Object 4 through 15	726
Custom Object 1 Child	727
Custom Web Applet	728
Delegated User	729
Exchange Rate History	729
Image	730
Interests	731
Invitee	732

Lead	734
Note	735
Opportunity	736
Opportunity Child	737
Products Detailed	737
Promotional Items Dropped	739
Recipient	740
Related Contact	741
Revenue	742
Sales Assessment	744
Sales Assessment Value	745
Sample Dropped	745
Service Request	747
Shared Address	748
Subbook	752
Team	753
User	754
User Default Book Per Type	755

## **Appendix A: Using Attachments with Web Services On Demand**

About Attachments	757
About The Attachment Element	757
Specifying File Attachment Content	759
Attaching Binary Data	759
Attaching a Text File	761
Attaching a File to Multiple Records	762
Retrieving or Querying File Attachments	764
Specifying URL Attachments	765

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

Mapping of Display Names	767
--------------------------	-----

## **Appendix C: Web Services On Demand Limiters**

Web Services On Demand Limiters	771
Web Services Request Rate	772

## **Index**

# 1

## What's New in This Release

### What's New in Oracle Web Services On Demand Guide, Version 24.0 (Oracle CRM On Demand Release 36)

Table 1 lists the changes in this version of the documentation to support Version 24.0 (Oracle CRM On Demand Release 36) of the software.

Table 1. What's New in Oracle Web Services On Demand Guide, Version 24.0 (Oracle CRM On Demand Release 36)

Topic	Description
<a href="#">"Outbound SSO" on page 65</a>	Modified topic. Information has been added to reflect that a hashed message authentication code (HMAC) token generated by Oracle CRM On Demand can now be used only once.
<a href="#">"Considerations when Rejecting Leads" on page 80</a> <a href="#">"Lead" on page 451</a>	New topic. For Web Services v1.0, the RejectReason field is now exposed for the Lead object.
<a href="#">"CurrentOrganizationRead" on page 264</a>	Modified topic. Two new company profile settings have been added: ExportRequestExpiryDays and ExportRequestAttachmentExpiryDays. These settings control the number of days export requests and export files are kept.
<a href="#">"ListRead" on page 309</a> <a href="#">"ListReadAll" on page 312</a>	New topics. A new Administrative Web service, Lists, is now available with methods to read configuration data for system lists, private lists, public lists, and role-specific lists. A new Manage Private Lists privilege has been added to allow access to all private lists in the company.
<a href="#">"PicklistRead" on page 322</a>	Modified topic. With the PicklistRead method you can now filter for active and inactive picklist values of a particular picklist.
<a href="#">"RelatedInformationLayoutUpsert" on page 333</a>	Modified topic. For the methods of the Related Information Administrative Web service, the Sequence field can now contain an integer with the range of values 0 through 19.
<a href="#">"RelatedInformationLayoutUpsert" on page 333</a>	Modified topic. For the Opportunities related record type for certain parent record types, you can now optionally sort the related information records by close date. The description of the <SortField> element has been updated accordingly.

## What's New in This Release ■

---

Table 1. What's New in Oracle Web Services On Demand Guide, Version 24.0 (Oracle CRM On Demand Release 36)

Topic	Description
<a href="#">"ReportFolderRead" on page 334</a> <a href="#">"ReportFolderReadAll" on page 335</a> <a href="#">"ReportFolderUpsert" on page 336</a>	Modified topics. For the methods of the Report Folders Administrative Web service, V3 dashboard folders are now supported.
<a href="#">"BulkOpImportCreateRequest" on page 366</a>	Modified topic. A new argument, SendUserLogFile, has been added for the BulkOpImportCreateRequest method. This argument specifies whether the user log file is attached to the email notification for bulk data import requests.



# 2

## Overview of Web Services On Demand

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

- [About Web Services on page 17](#)
- [Oracle CRM On Demand Web Services and Integration with Oracle CRM On Demand on page 18](#)
- [Web Services and the Oracle CRM On Demand Objects on page 21](#)
- [Field Types Supported by Oracle CRM On Demand on page 26](#)
- [Web Services Utilization on page 40](#)
- [Web Service Client Name Identification on page 41](#)
- [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.

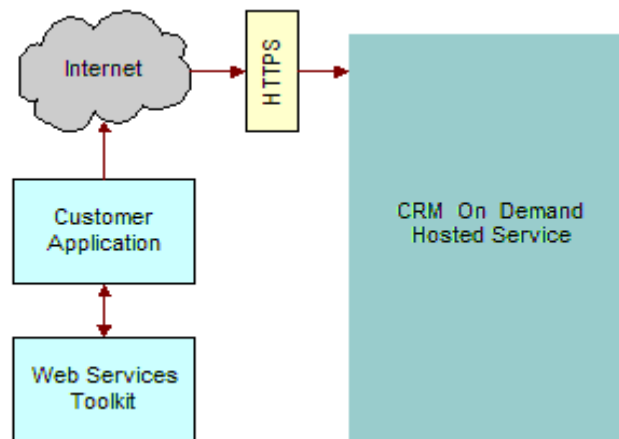


Figure 1. How Web Services Communicate with Oracle CRM On Demand

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 20](#).
- 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 55](#).

### 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 56](#).

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. Also used to access custom Web applets.

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.

## Overview of Web Services On Demand ■ Web Services and the Oracle CRM On Demand Objects

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
<p>Returns all child records even if the condition is true for one child.</p> <p>For example, the QueryPage call returns all partner children from an account even if the condition is true for only one partner child</p>	<p>Outputs only the specific child whose condition was met.</p> <p>For example, QueryPage returns only the specific partner child from the account for which the condition was true.</p>
<p>UseChildAnd argument of QueryPage call is available for using OR/AND logic between parent and child</p>	<p>The UseChildAnd argument is not available.</p> <p>Instead, by default, all parent records matching the parent criteria and only children matching the child criteria are returned.</p>
<p>Operators cannot be used to construct complex queries across multiple fields</p>	<p>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].</p>
<p>Sort order is not customizable</p>	<p>Sortorder and sortsequence arguments are available to customize the sorting order of the records</p>
<p>Update call removes child objects not specified in the request</p>	<p>An Execute call with "operation=update" at the parent level removes the unspecified children in the request</p>
<p>InsertChild call is used to insert the children for existing parent objects</p>	<p>For Web Services v2.0:</p> <ul style="list-style-type: none"> <li>■ Insert call can be used to insert both parent records and child records.</li> <li>■ If a child node is specified in the request, the Insert call inserts the child and associates it with the existing parent record.</li> <li>■ If a child node is missing, the Insert call inserts only the new parent record.</li> </ul>
<p>UpdateChild call is used to update child records</p>	<p>For Web Services v2.0:</p> <ul style="list-style-type: none"> <li>■ Update call can be used to update parent records and child records</li> <li>■ If a child node is specified in the request, the Update call updates the child in the existing parent record</li> <li>■ If the child node is missing, the Insert call updates only the existing parent record</li> </ul>

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"> <li>■ Delete call can be used to delete both parent records and child records</li> <li>■ 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.</li> <li>■ If the child node is missing, the Delete call deletes the existing parent record.</li> </ul>
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.
Does not support access to custom Web applets.	Supports access to custom Web applets as read-only child objects of a parent object.

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 26](#).
- 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.

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`.

The following procedure describes how to view or modify the integration tag information:

### *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 using the following procedure:

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

- 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 as required.
- 3 From the Document list, select WSDL.
- 4 From the Type list, select Custom.
- 5 From the WSDL Object list, select the required record type.
- 6 From the Select Related Information list, select the child record types that you wish to include in the WSDL.
- 7 Click Download.
- 8 Save the WSDL file to your computer.

For more information about downloading WSDL files, see ["Downloading WSDL Files" on page 51](#).

## 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 5550199
  - 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>&gt;'01/01/2004 00:00:00'</CloseDate> you get an error, because the time parameter 00:00:00 is only valid for fields of type Date/Time and not for fields of type Date.

## Web Services v2.0

The Web Services v2.0 API supports strong data 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
UTC_DATETIME	xsd:dateTime
ID	xsd:string
NOTE	xsd:string
PHONE	xsd:string
TEXT	xsd:string
INTEGER	xsd:int

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

Data Type	Mapped XSD Data Type
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 v1.0, the time zone is assumed to be the logged in user's time zone, which is determined from the user's locale.

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)?

An example of a date and time in this format is:

2002-10-10T12:00:00-05:00

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

The same date and time for UCT, which is equivalent to the GMT time zone is as follows:

```
2002-10-10T17:00:00Z
```

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: @riqhtequip.com
- No at sign (@) character, for example: isampleriqhtequip.com
- No period (.) character, for example: isample@riqhtequipcom
- No domain, for example: isample@
- No domain suffix such as com, for example: isample@riqhtequip
- Multiple at signs (@), for example: isample@@riqhtequip.com
- Consecutive period (.) characters, for example: isample@riqhtequip..com
- Spaces in the string, for example: isa mple@riqhtequip
- Characters other than the following in the local part of an email address:
  - Uppercase and lowercase letters (case insensitive)
  - The digits 0 through 9
  - The characters:
    - Exclamation point (!)
    - Hash symbol (#)
    - Dollar sign (\$)

- ❑ Percent (%)
  - ❑ Ampersand (&)
  - ❑ Single quotation sign (')
  - ❑ Asterisk (\*)
  - ❑ Plus sign (+)
  - ❑ Minus sign (-)
  - ❑ Slash (/)
  - ❑ Equal sign (=)
  - ❑ Question mark (?)
  - ❑ Caret (^)
  - ❑ Underscore (\_)
  - ❑ Back single quotation mark (`)
  - ❑ Left curly brace ({)
  - ❑ Vertical bar (|)
  - ❑ Right curly brace (})
  - ❑ Tilde (~)
- Any special characters in the domain name of an email address. These special characters are the same as those allowed in the local part of the email address, and also the left and right parentheses ().

## Unicode Characters in Email Addresses

For some fields in Oracle CRM On Demand, email addresses can include most Unicode (UTF-8) characters, if the Allow Unicode Characters in Email Fields company profile setting is selected. This allows, for example, email addresses to contain accented characters.

Oracle servers do not support Unicode characters in email addresses, therefore such addresses are not allowed in User email fields. However, Oracle Eloqua Marketing Cloud Service does support Unicode characters, therefore you can save email addresses containing Unicode characters in Contact and Lead email fields and use the Send Email via Engage button on Contact Detail, Contact List, Lead Detail, or Lead List pages to send the emails. For more information about the use of Unicode characters in email addresses, see *Oracle CRM On Demand Online Help*.

## 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](#) show 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 36.

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 <a href="#">Table 7</a>
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 <a href="#">Table 7</a>
County	County	Bill To County	Ship To County	Personal County	Primary County	Group C See <a href="#">Table 7</a>
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 <a href="#">Table 6</a> .



Table 7. Groups of Countries with Different Address Field Mappings

Group	Countries
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>
```

```
<Integrati onI d>10A2-R7C30</Integrati onI d>
  <Address>100 Mai n Street</Address>
  <StreetAddress2></StreetAddress2>
  <Pri maryAddressLi ne1>100 Mai n Street</Pri maryAddressLi ne1>
</Address>
...
</Li stOfAddress>
</Account>
</Li stOfAccount>
</ns: AccountQueryPage_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envel ope>
```

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

```
<?xml versi on="1. 0" encodi ng="UTF-8"?>
```

```
<SOAP-ENV: Envel ope xml ns: SOAP-ENV="http: //schemas. xml soap. org/soap/envel ope/"
xml ns: xsi ="http: //www. w3. org/2001/XMLSchema-i nstance" xml ns: xsd="http: //www. w3. org/
2001/XMLSchema">
```

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

```
<Li stOfAccount xml ns="urn: /crmondemand/xml /Account/Data" l astpage="true">
```

```
<Account>
```

```
<Locati on>Togo</Locati on>
```

```
<AccountName>ACCOUNTTEST2</AccountName>
```

```
<Li stOfAddress l astpage="true">
```

```
<Address>
```

```
<I d>10A2-R7I MS</I d>
```

```
<StreetAddress3></StreetAddress3>
```

```
<Country>Togo</Country>
```

```
<County>10222</County>
```

```
<Descri pti on></Descri pti on>
```

```
<Provi nce></Provi nce>
```

```

    <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 as shown in [Table 9](#).

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

## Support for Web Link Fields

A Web link field is a custom field that allows you to place a hyperlink to an external Web site or Web-based application in a record in Oracle CRM On Demand. You can access Web link fields through Web Services v2.0 calls to retrieve the URL and the display text for the Web link field. You cannot create, update, or delete Web links through Web service calls.

Web services calls retrieve custom Web link field values in anchor tags with the following format:

```
<a href="www.Link.com">Web Link Display Text</a>
```

However, in responses the URL and display text are encoded as shown in the following example:

```
<CustomWebLink0>&lt; a
href="https&#x3a; &#x2f; &#x2f; www. google. com&#x2f; &#x3f; gws_rd&#x3d; ssl &#x23; q&#x3d; Oracl e&#x25; 20Corporati on" &#x26; Fi nd Oracl e
Corporati on&#x2f; /a&#x26; </CustomWebLink0>
```

In the href attribute value, the URL is HTML attribute value encoded. The anchor tag content corresponds to the Web link display text. The URL and display text must be extracted from the encoded values, as described in the following section.

**NOTE:** Internally, concatenated fields are similar to Web link fields but have only display text and do not have a URL. Responses to queries for concatenated fields do not contain anchor tags; the evaluated display text is retrieved as the field value. For more information about concatenated fields, see ["Support for Concatenated Fields" on page 39](#).

## Example of Retrieving and Extracting Web Link Values

As an example, a contact can have a Web link field, CustomWebLink0, with the following configuration:

- **Display Text.** Find %%%First\_Name%%% %%%Last\_Name%%%
- **URL.** https://www.google.com/?gws\_rd=ssl#q=%%%First\_Name%%% %%%Last\_Name%%%

For information about how to set up Web links in the UI, see *Oracle CRM On Demand Online Help*.

For a Web service QueryPage call for a Contact having Id = 10A2-230MBG, First Name = Oracl e, and Last Name = Corporati on, the request and response are shown in ["Sample SOAP Request for a Custom Web Link Field" on page 38](#) and ["Sample SOAP Response for a Custom Web Link Field" on page 38](#).

The CustomWebLink0 field value is retrieved as:

```
<CustomWebLi nk0>&lt; a
href="https&#x3a; &#x2f; &#x2f; www. googl e. com&#x2f; &#x3f; gws_rd&#x3d; ssl &#x23; q&#x3d; Oracl e&#x25; 20Corporati on" &#x26; Fi nd Oracl e
Corporati on&#x2f; /a&#x26; </CustomWebLi nk0>
```

The display text and URL must then be extracted from the encoded values in the Web service response.

### *To extract Web link display text*

- From the field value, extract the Web link display text using the regular expression pattern "`<a.*?>(.*?)</a>`".

For the CustomWebLink0 example, the Web link display text is extracted as:

```
Fi nd Oracl e Corporati on
```

### *To extract a Web link URL*

- 1 From the field value, extract the Web link URL using the regular expression pattern "href=\"(.\*)\"".

For the CustomWebLink0 example, the following is extracted:

```
https&#x3a; &#x2f; &#x2f; www. googl e. com&#x2f; &#x3f; gws_rd&#x3d; ssl &#x23; q&#x3d; Oracl e&#x25; 20Corporati on
```

- 2 The URL is HTML attribute value encoded, so to extract the actual URL, perform an unescape HTML operation.

For the CustomWebLink0 example, the following URL is extracted:

```
https: //www. googl e. com/?gws_rd=ssl #q=0racl e%20Corporati on
```

### Sample SOAP Request for a Custom Web Link Field

The following is an example of a query request for the Web link field, CustomWebLink0:

```
<S: Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<SOAP-ENV: Header/>
```

```
<S: Body>
```

```
<ns2: ContactQueryPage_Input xmlns="urn:/crmondemand/xml/Contact/Data"
xmlns:ns2="urn:crmondemand/ws/ecbs/contact/" xmlns:ns3="urn:/crmondemand/xml/
Contact/Query">
```

```
<ns3: ListOfContact pagesize="1">
```

```
<ns3: Contact>
```

```
<ns3: CustomWebLink0/>
```

```
<ns3: Id>='10A2-230MBG' </ns3: Id>
```

```
</ns3: Contact>
```

```
</ns3: ListOfContact>
```

```
</ns2: ContactQueryPage_Input>
```

```
</S: Body>
```

```
</S: Envelope>
```

### Sample SOAP Response for a Custom Web Link Field

The following response contains the encoded URL and display text for the Web link field, CustomWebLink0:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns:ContactQueryPage_Output xmlns:ns="urn:crmondemand/ws/ecbs/contact/">
      <ListOfContact xmlns="urn:/crmondemand/xml/Contact/Data" Lastpage="true">
        <Contact>
          <Id>10A2-230MBG</Id>
          <CustomWebLink0>&lt;t; a
href="https&#x3a; &#x2f; &#x2f; www. googl e. com&#x2f; &#x3f; g
ws_rd&#x3d; ssl &#x23; q&#x3d; Oracl e&#x25; 20Corporati on"&gt; Fi n
d Oracle Corporation&lt;t; /a&gt; </CustomWebLink0>
        </Contact>
      </ListOfContact>
    </ns:ContactQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## Support for Concatenated Fields

A concatenated field is a field that can display the values from multiple fields and can also display additional text.

You cannot use Web services calls to update or query values within a concatenated field directly. To update or query the values of a concatenated field through Web service calls, you must update or query each of the individual fields separately. When you perform a QueryPage call on a concatenated field, the display text for the concatenated field is returned in the response.

For more information about concatenated fields, see *Oracle CRM On Demand Online Help*.

The Concatenated Field Administrative Service allows you to query, insert, and update concatenated field configuration data. For more information, see [“ConcatenatedFieldRead” on page 258](#), [“ConcatenatedFieldReadAll” on page 259](#), and [“ConcatenatedFieldUpsert” on page 260](#).

## Support for Maskable Fields

A maskable field is a field in which some of the data can be hidden from view from some users. The administrator can set up some custom fields as maskable fields for certain record types.

Users whose role includes the View Masked Data privilege, can view all of the data in a maskable field. However, users whose role does not include the View Masked Data privilege can see only the last four characters of the value in maskable fields. All of the other characters in the field are represented by the characters XXXX.

For example, if a maskable field contains the value 102030456789, then you see the following:

XXXX6789

If your user role includes the View Masked Data privilege, then the following applies to Web services requests:

- You can insert data into maskable fields and update maskable fields.
- The full (unmasked) field value is returned in query results.

If your user role does not include the View Masked Data privilege, then the following applies to Web services requests:

- You can insert data into maskable fields and update maskable fields.
- The masked field value is returned in query results.
- You cannot use maskable fields in the search specification or sort specification in queries. Such queries return an error message.

**NOTE:** If a maskable field is set up as read-only for the record type, or for the page layout that is assigned to a user's role for the record type, then you cannot update the field.

For more information about maskable fields, see *Oracle CRM On Demand Online Help*.

## 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. For more information about the Web Service Client Name parameter, see [“Web Service Client Name Identification” on page 41](#).
- **# 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.
- **Type.** The user agent value for the request. For client integrations other than Oracle client integrations, this value defaults to Web Services. For Web services requests, language-independent codes are used instead of the display values used in the Oracle CRM On Demand UI, as shown in the following table:

Display Value for Type	Language-independent Code
Interactive	UI
Unknown	UNK
Web Services	WS

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 229](#).

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 `Inval id Cl ient Name` is displayed in the Web Services Utilization page. The <ClientName> value is restricted to 100 characters; for any value longer than 100 characters, `Inval id Cl ient Name` is displayed in the Web Services Utilization page.

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

[Devel oper], [Cl ient Name]

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

XYZ Consul ti ng, Account Synchroni zati on Uti li ty

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, as follows:

- If a stateless request execution results in explicit login, then 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, then 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, the following considerations apply:

- If the stateful request requires tracking of the client name, then it must be specified in the stateful login operation.
- If a client name is specified in a stateful request execution, then 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 stateful and 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 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 569](#)). 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 540](#))
- The Export Assistant
- The List Management Export feature in the Service Allotment Usage History page under Company Administration.

For information about the Export Assistant and the List Management Export feature, see *Oracle CRM On Demand Online Help*.

## 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 79.

## 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
<b>Nonquery operations</b>	
Account insert request containing a single Account record (with no child operations)	1
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

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

Type of Request	Number of Operations
<b>Query operations</b>	
Simple query for a set of Accounts	1
Query for a set of Accounts and the associated Contacts for each Account	1 + <i>n</i> , where: <ul style="list-style-type: none"><li>■ 1 operation to retrieve the set of <i>n</i> Accounts matching the specified filter criteria</li><li>■ <i>n</i> operations to retrieve the set of Contacts associated with each Account (1 operation for each Account)</li></ul>

### 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 53](#)

### 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.

### Web Services Privileges

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 20](#).

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.

### Downloading WSDL Files

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

- **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 you can also download Custom or Generic Schema files, as described in [Downloading Schema Files](#).

### Downloading WSDL Files for Web Services v1.0 and Web Services v2.0

You can download a WSDL file for each record type and you can choose the child record types to include in the WSDL file for the parent record type.

#### *To download a WSDL file for Web Services v1.0 and Web Services v2.0*

- 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 as required.
- 3 From the Document list, select WSDL.
- 4 From the Type list, select Custom or Generic.
- 5 From the WSDL Object list, select the required record type, for example, Account.

The objects displayed in the WSDL Object list depend on the record types that are set up for your company. All of the child record types for the selected record type are displayed, in alphabetical order, in the Available Related Information list, regardless of whether you have access to the child record types. For Web Services v2.0, CustomWebApplet is also displayed.

- 6 Move the child record types that you wish to include in the WSDL from the Available Related Information list to the Selected Related Information list.
- 7 Click Download.
- 8 Save the WSDL file to your computer.

### Downloading WSDL Files for Service APIs

You can download a WSDL file for each of the service APIs. If you download the WSDL for Integration Event, then you must additionally download the Schema files for each of the record types for which you wish to generate integration events, see [“Downloading Schema Files” on page 52](#).

#### *To download a WSDL file for Service APIs*

- 1 Navigate to the Web Services Administration page.
- 2 From the Select Service drop-down list, select Service APIs.  
**NOTE:** If you select the Merge Service API, you can select Custom or Generic from the Type list and download a Schema file for this Service API.
- 3 From the Type list, select Custom or Generic.
- 4 From the WSDL Object list, select the required name of the Service API.
- 5 Click Download.
- 6 Save the WSDL file to your computer.

### Downloading WSDL Files for Administrative Services

You can download a WSDL file for each Administrative service.

**NOTE:** To download the Assessments Scripts WSDL or the Access Profile WSDL, your user role must include the Manage Roles and Access privilege.

#### *To download a WSDL file for Administrative Services*

- 1 Navigate to the Web Services Administration page.
- 2 From the Select Service drop-down list, select Administrative Services.
- 3 From the WSDL Object list, select the required service.
- 4 Click Download.
- 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 API.

If you create custom fields or rename fields for a record type, then you cannot use the Generic XSD files for tracking these fields. Instead, you must download a Custom XSD file using the Custom option on 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 181](#).

*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 Document list, select Schema.
- 4 From the Type list, select Custom or Generic.

Depending on whether you select Custom or Generic, custom fields are displayed differently in the Schema file. For Custom Schema, the XML tags for the custom fields are based on the integration tags from Field Setup. For Generic Schema, the custom fields are based on generic XML tags.

- 5 From the WSDL Object list, select the required record type, for example, Account.

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

- 6 From the Select Related Information list, select the child record types that you wish to include in the Schema.
- 7 Click Download.
- 8 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 73](#).



# 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 55](#)
- [Using Stateless Web Service Requests on page 56](#)
- [Using Stateful Web Services Requests on page 59](#)
- [Integration Requests for the Web Services Session on page 63](#)
- [Comparison of Stateless Versus Stateful Web Services Transactions on page 64](#)
- [Single Sign-On on page 65](#)

### 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 56](#)), 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
- SSO with SAML v1.1

## 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 65](#).

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

```
<soap: Header>
```

```
  <wsse: Security>
    <wsse: SecurityTokenReference>
      <wsse: KeyIdentifier ValueType="http://schemas.crmondemand.com/ws/2011/01/secext#SS0TokenKeyIdentifier">$6$qx6pJ/czNw01trwQRazQ26j4osNiQHMoqQSwRfpz/6HX2D5cw=;$6$IjwK0/BBobW5oiuqC7P/TxwOBX1LxVpExR9vp7P5J/ki xzGFWIj 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 65](#).

### SSO with SAML v1.1 or v2.0

This login mechanism is a type of inbound SSO, see [“Inbound SSO” on page 65](#).

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 58](#).

**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.

For stateless requests where the <SessionKeepAlive> parameter is set to false, where it is empty, or where <SessionKeepAlive> is not present, the Java session will be logged off by Oracle CRM On Demand after the request has been processed.

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: SessionKeepAlive xmlns:crmod="urn:crmondemand/ws">true
</crmod: SessionKeepAlive>
  <wsse: Security>
    <wsse: SecurityTokenReference>
      <wsse: KeyIdentifier ValueType="http://schemas.crmondemand.com/ws/2011/01/secext#SS0TokenKeyIdentifier">$6$qx6pJ/rGFI DyJBy91C3Uxi fc3oDEg/
      Sshtrsyj 2z27RoHm05M=; $6$j Wkqu/
      Hvl Xoqw2FoHx10GDcYDc5WdZ2UV4v1G06AMEBNbWZv0JZi nyJ1C4v70HJR400VhZ4J3X+U/
      EvrP1c82LYHV0HA==</wsse: KeyIdentifier>
    </wsse: SecurityTokenReference>
  </wsse: Security>
</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 by including the session ID 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. Attaching the session ID as a parameter to the URL request line is contrary to security standards and is not good practice.

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 63](#).

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

All requests must use HTTP Secure (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 Single 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](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 for stateless requests is used, but with a WSSE draft namespace (<http://schemas.xmlsoap.org/ws/2002/04/secext> or <http://schemas.xmlsoap.org/ws/2002/07/secext>) instead of the WSSE Version 1.0 Namespace. For more information, see ["Login with UserName and Password in the SOAP Security Header" on page 57](#).

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="wsse: 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 65](#).

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 65](#).

### 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 65](#)

## 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.

# 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 the following:

- 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 set to true 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 and can be used only once. Any subsequent attempts to use the same token for validation or to obtain a session ID will fail.

### 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/Integrati on?command=ssoi turl &ssoi d=<SSO_Company_I denti fi er>
```

- 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](https://codesamples.samplecode.oracle.com/servlets/Scarab/action/ExecuteQuery?query=crm_on_demand)

# 5

## Best Practices for Designing Client Application

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 67](#)
- [Best Practices for Integration Performance on page 80](#)
- [Best Practices for Integration Management on page 86](#)

### 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 207](#).

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 81](#).

## 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. For more information, see [“Logging of SOAP Requests and Responses” on page 71](#).

- 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*.

## Logging of SOAP Requests and Responses

SOAP requests and SOAP responses are important information that can assist you in troubleshooting your integration when issues arise, as they capture exactly what was sent to and received from Oracle CRM On Demand. Logging the requests and responses can also help you better understand your integration.

It is a best practice and strongly recommended that your integration logs any SOAP request sent to Oracle CRM On Demand Web Services and any SOAP response received for easier troubleshooting of your integration. It is also beneficial to log the SOAP requests and responses when an issue arises that requires a service request to be created by technical support. The logs with the SOAP requests and responses can help the Oracle CRM On Demand team with reproducing and identifying the issue.

There are different ways in which you can log SOAP requests and responses, for example:

- Depending on the technology that you use to implement your client application, your client application can output the SOAP requests and responses to log files when they are sent and received in the application. For example, with Java API for XML Web Services (JAX-WS) this can be done by implementing SOAP message handlers. With ASP.NET it can be implemented by using the SOAP extensions framework.
- You can use proxies to record any HTTP traffic between the client and Oracle CRM On Demand.

## 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.

## Understanding Web Services Transactions

When designing client applications, remember that Web service requests can be treated as transactions. A transaction succeeds if all operations in the transaction are successful, otherwise the transaction is rolled back.

For Web Services On Demand calls, the transaction boundary is at message level. That is, if a Web service request is received with 10 operations (for example, 10 inserts), and if the 10th operation fails for some reason, then all of the 9 previous successful operations are rolled back and the initial state is restored.

A transaction is created only when there are operations on more than one record. Transactions are supported for both Web Service v1.0 and Web Services v2.0 calls.

## 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 73.
- 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"`, and do the following:
  - 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"` and do the following:
  - 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](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.getModId() != null) {
    myns3_int__java_lang_Integer_Int_Serializ
er.setNullable(false);
    myns3_int__java_lang_Integer_Int_Serializ
er.serialize(instance.getModId(),
    ns2_ModId_QNAME, null, writer, context);
}
...
}

```

Figure 3. A doSerialize() Method After Splitting

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

```

{
    // @GeneratedBlockBegin (value={"oracle.j2ee.ws"});
    activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance =
    (activity.proxy.types.crmondemand.xml.activity.data.ActivityData)obj;

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

public activity.proxy.types.crmondemand.xml.activity.data.ActivityData
doSerialize1(activity.proxy.types.crmondemand.xml.activity.data.ActivityData instance, XMLWriter writer,
SOAPSerializationContext context) throws Exception
{
    if (instance.getModifiedDate() != null) {
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializ
er.setNullable(false);
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializ
er.serialize(instance.getModifiedDate()
,
        ns2_ModifiedDate_QNAME, null, writer, context);
    }
    if (instance.getCreatedDate() != null) {
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializ
er.setNullable(false);
        myns3_dateTime__java_util_Calendar_DateTimeCalendar_Serializ
er.serialize(instance.getCreatedDate(),
        ns2_CreatedDate_QNAME, null, writer, context);
    }
    ...
return instance;
}

```

```

public activity.proxy.types.crmdemand.xml.activity.data.ActivityData
doSerialize2(activity.proxy.types.crmdemand.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
<b>Input through Web service</b>		
14045550199123	1 (140) 555-0199#1123	+1 404 5550199 #123
140455501991	1 (140) 555-0199#1	+1 404 5550199 #1

Table 14. Formatting of Telephone Number Values Entered Through Web Services or the UI

Input Mechanism	Formatting in UI	Formatting in Web Service Response
+14045550199,123	1 (4045550199) 123	+1 4045550199 123
<b>Input through UI</b>		
14045550199123	14045550199123	+1 4045550199123
14045550199	1 (404) 555-0199	+1 404 5550199
+14045550199,123	1 (404) 555-0199#123	+1 404 5550199 #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) 555-0199 123
- 1 (404) 555-0199 #123
- 1 (140) 555-0199#1
- 1 (140) 555-0199

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) 555-0199 #123</WorkPhone>
    <PHONE_000>1 (404) 555-0199 #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) 555-0199 #123</WorkPhone>
  <PHONE_000>1 (404) 555-0199 #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) 555-0199 #123
```

is displayed as the following in the Web service response:

```
+1 404 5550199 #123
```

and as the following in the UI:

```
1 (404) 555-0199 #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:

```
+1 404 555-0199 #123
```

is displayed as the following in the Web service response:

```
1 404 5550199 #123
```

and as the following in the UI:

```
1 (404) 555-0199 #123
```

There are locale considerations when entering telephone numbers. As an example, if the user locale is set to Australia (country code +61) and a Web service request contains the 10 digit telephone number: 0477777777, then the number displayed in the UI is:

```
+1 047 777-7777
```

as the number displayed defaults to the North American Numbering Plan if the country code is not included. In this case, the user locale (Australia) is not considered.

If you enter the same telephone number (0477777777) in the UI or through import, the telephone number displayed in the UI is:

```
(04) 7777 7777
```

where the country code is set to Australia.

To make the behavior consistent across Web service requests and the UI, it is recommended that in Web service requests you prefix the number with the + character and country code. For the previous example, the telephone number in the Web service request would be:

+61477777777

Telephone numbers submitted through Web service requests must not begin with an alphabetic character and must either begin with a numeric value or a + character. 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*.

## Considerations when Rejecting Leads

In the Oracle CRM On Demand UI you can reject a lead by selecting the Reject button on a lead detail page. The status of the lead then changes to Rejected and you must select a value in the Reject Code field and you can enter a value in the Reject Reason field.

For Web Services v1.0 and Web Services v2.0 you can make calls to update the RejectCode and RejectReason fields for the Lead object; however, there is no validation of the Status field for such calls. Therefore, when designing your client applications you must add your own validations to ensure that the Status, RejectReason, and ReasonCode fields are set correctly.

For more information about working with leads, including rejecting leads and lead qualification, 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 68](#). 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 55](#).

## 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 762](#). 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 200](#) and [“UserUsageQueryPage” on page 229](#).

## Using the QueryPage Method

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

- If you intend to paginate through the snapshot of data returned by `QueryPage`, it is recommended to use a stateful Web services request. Using a stateless Web services request is not recommended because it can return the same record in multiple pages as you paginate though the results returned.
- If you do use stateless `QueryPage` operations and then paginate through the snapshot of data returned, it is recommended that the `QueryPage` request uses sort criteria.
- 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 85](#).
- 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.
- Avoid filtering on a calculated field because it impacts performance.
- 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 96](#), or [“QueryPage” on page 123](#).

### Querying Records with Foreign Key fields with the No Match Row Id Value

The No Match Row Id value for a foreign key field indicates one of the following:

- The foreign key refers to a nonexistent record.
- The foreign key value is empty.

There are different cases for which No Match Row Id is set for a foreign key field. For example, if you create a record without specifying a foreign key field in the insert operation, then the foreign key value is empty and is therefore set to No Match Row Id. In this case, querying the record returns `<ForeignKeyId>No Match Row Id</ForeignKeyId>`.

However, if you update a record and specify blank as the foreign key value, then it is set to empty. In this case, querying the record returns `<ForeignKeyId></ForeignKeyId>`.

You must take the No Match Row Id value into consideration for foreign key fields when querying records. If you try to retrieve all records where the foreign key value is not null, then the query returns records including those with the No Match Row Id value. Therefore, when querying a foreign key, the query criteria must include checking for null or not null as well as the No Match Row Id value.

For example, when querying for accounts with primary contact associated to the account, the search criteria must specify that the PrimaryContactId is not null and not equal to No Match Row Id, as shown in the following code sample:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/
2001/XMLSchema">
  <soap:Body>
    <AccountQueryPage_Input xmlns="urn:crmondemand/ws/ecbs/account/10/2004">
      <ListOfAccount>
        <Account searchspec="[PrimaryContactId] IS NOT NULL AND [PrimaryContactId]
&lt;&gt; 'No Match Row Id' ">
          <AccountName />
        </Account>
      </ListOfAccount>
    </AccountQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

```
<PrimaryContactId></PrimaryContactId>
</Account>
</ListOfAccount>
</AccountQueryPage_Input>
</soap: Body>
</soap: Envelope>
```

## 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 the export and import functionality of Oracle CRM On Demand 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 173](#), and [DeleteEvents on page 163](#) 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 71](#).

## 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 describes the Web Services On Demand API Calls. It contains the following topics:

- [Web Services On Demand API Calls on page 89](#)
- [Web Services v1.0 API Calls on page 90](#)
- [Web Services v2.0 API Calls on page 107](#)
- [Service API Calls on page 147](#)
- [Administrative Services API Calls on page 233](#)
- [Data Loader API Calls on page 366](#)

## Web Services On Demand API Calls

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

- 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](https://codesamples.samplecode.oracle.com/servlets/tracking/action/ExecuteQuery?query=crm_on_demand).

## Considerations When Deleting Child Records

When you delete a child record, either through the UI or a Web service call, a foreign key reference to the child is maintained in the parent record. This foreign key reference is maintained so that if the deleted child record is restored, the association that the record had with parent records can also be restored.

Web service calls querying the record with the foreign key reference can give unexpected results. To avoid this situation, it is recommended that before you delete a child record, you first remove the association between the child record and the parent record. Web service calls will then not return the foreign key reference.

If you want to remove a retained foreign key reference from a parent record, associate a temporary child record, then remove the association and delete the temporary child record.

## 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.

**NOTE:** For Web Services v1.0, operations on a child object with more than 10,000 records cannot be processed. It is recommended that you use Web Services v2.0 operations to overcome this limitation.

## 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 the business logic of Oracle CRM On Demand, 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.

Cascade delete of child records is consistent with the cascade delete functionality in the Oracle CRM On Demand UI. It is recommended that deletion of parent objects through Web services requests also specify which child records are to be deleted to ensure the appropriate deletion of child records. For more information about deleting records, see *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 26](#).

## Arguments

[Table 16](#) describes the arguments taken by the Delete method.

Table 16. 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.

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 17 describes the arguments taken by the DeleteChild method.

Table 17. 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.

Insert calls must include values for fields that are required for the object. Fields can be conditionally required, that is they become required if the data specified for other fields on the record meets the criteria defined in an expression for the field. For more information about conditionally-required fields, see *Oracle CRM On Demand Online Help*.

Table 18 outlines how the Insert method acts on parent and child objects to create or update an object instance.

Table 18. 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 19 describes the arguments taken by the Insert method.

Table 19. 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 20 describes the arguments taken by the InsertChild method.

Table 20. 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.

Insert calls must include values for fields that are required for the object. Fields can be conditionally required, that is they become required if the data specified for other fields on the record meets the criteria defined in an expression for the field. For more information about conditionally-required fields, see *Oracle CRM On Demand Online Help*.

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

Table 21. 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 22 describes the arguments taken by the InsertOrUpdate method.

Table 22. 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>
<PageSize>20</PageSize>
<ListOfAccount>
```

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

- If the number of records in the record set is less than the PageSize value, the full record set is returned, and the LastPage argument is set to true.
- If the number of records in the record set exceeds the PageSize value, only the number of records specified by the PageSize parameter is returned, and the LastPage value is false.

- For a query whose record set exceeds the PageSize value, incrementing the StartRowNum argument by PageSize returns the next PageSize number of records. For example, to return a record set with 997 records you can use a PageSize of 100 as follows:

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

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

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

## About Oracle CRM On Demand Query Syntax

The query syntax supports only a small subset of binary and unary operators. No Siebel Query Language constructs or functions are supported. The query syntax is summarized in [Table 23](#).

Table 23. Query Syntax for QueryPage

Syntax Type	Notes
<b>expression</b>	
{ <i>Operator</i> } { <i>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>expression</i> ) <i>conjunction</i> ( <i>expression</i> )	A conjugated expression must be enclosed in parentheses to avoid ambiguity. However, nonconjugated expressions must not be enclosed in parentheses.
<b>conjunction</b>	
OR	None
AND	None
<b>unary operator</b>	
IS NULL	Used to find a match for a value that has no value
<b>binary operator</b>	
=	None
~=	Denotes a case-insensitive exact search (no wildcards used)
<	Must be specified as &lt; to ensure well-formed XML.
<=	Must be specified as &lt; =
>	None
>=	None
<>	Must be specified as &lt; >
LIKE	Wildcard characters are treated as such only in the context of the operator LIKE.
~LIKE	Denotes a case-insensitive wildcard search

Table 23. Query Syntax for QueryPage

Syntax Type	Notes
value	
'literal'	<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. In some cases, you can however use shadow fields, see [“Using Shadow Fields for Better Search Performance”](#) on page 130.

## Querying Oracle CRM On Demand Data Using Web Services

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

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

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

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

Table 24 specifies the returned record sets for various values of each field value that maps to the preceding list.

Table 24. 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 24. 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.  <b>NOTE:</b> 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 24. 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 25 describes the arguments taken by the QueryPage method.

Table 25. 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 PageSize=100 and 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 26 outlines how the Update method acts on parent and child objects to update an object instance.

Table 26. 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 27 describes the arguments taken by the Update method.

Table 27. 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 28](#) describes the arguments taken by the UpdateChild method.

Table 28. 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 29 on page 108](#). 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 29. 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:

- **ManagerBooks.** Provides access to the same set of records that are returned when Manager Visibility is enabled in the company profile and ViewMode is set to AllBooks.
- **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.

## The Field-level Replace Operation

For the Web Services v2.0 Update and Execute method calls you can specify the operation attribute with the value "replace" on reference ID fields. Requests that include this attribute remove the existing association between the parent and the referenced record and create a new association with a specified record. An example of such a request is as follows:

```
<CampaignUpdate_Input xmlns="urn:crmondemand/ws/ecbs/Campaign/">
  <ListOfCampaign>
    <Campaign>
      <Id>1-FRW01</Id>
      <CustomObject1Id operation="replace">10A2-LVL4X</CustomObject1Id>
    </Campaign>
  </ListOfCampaign>
</CampaignUpdate_Input>
```

In this example, the existing association between the campaign parent and the custom object child record is replaced with an association between the campaign parent and the custom object record with the ID: 10A2-LVL4X.

The list of objects and reference ID fields that support the replace operation in the current release is shown in [Table 30](#).

Table 30. Objects and Fields That Support the Replace Operation

Object	Reference ID Fields
Account	CustomObject1Id, CustomObject2Id, CustomObject3Id, PrimaryContactId
Activity	CustomObject1Id, CustomObject2Id, CustomObject3Id, PrimaryContactId
Campaign	CustomObject1Id, CustomObject2Id, CustomObject3Id
Contact	CustomObject1Id, CustomObject2Id, CustomObject3Id

Table 30. Objects and Fields That Support the Replace Operation

Object	Reference ID Fields
Custom Object 1, Custom Object 2, Custom Object 3	AccountId, ActivityId, AssetId, CampaignId, ContactId, CustomObject1Id, CustomObject2Id, CustomObject3Id, DealerId, FundId, FundRequestId, HouseholdId, LeadId, MedEdId, OpportunityId, PortfolioId, ProductId, ServiceRequestId, SolutionId, VehicleId
Custom Object 1 child, Custom Object 2 child, Custom Object 3 child	AccountId, CampaignId, ContactId, CustomObject1Id, CustomObject2Id, CustomObject3Id, DealerId, FundId, HouseholdId, LeadId, MedEdId, OpportunityId, PortfolioId, ProductId, ServiceRequestId, SolutionId, VehicleId
Dealer	CustomObject1Id, CustomObject2Id, CustomObject3Id
Fund	CustomObject1Id, CustomObject2Id, CustomObject3Id
Fund Request	CustomObject1Id, CustomObject2Id, CustomObject3Id
Household	CustomObject1Id, CustomObject2Id, CustomObject3Id
Lead	CustomObject1Id, CustomObject2Id, CustomObject3Id
MedEdEvent	CustomObject1Id, CustomObject2Id, CustomObject3Id
Opportunity	CustomObject1Id, CustomObject2Id, CustomObject3Id
Partner	CustomObject1Id, CustomObject2Id, CustomObject3Id
Portfolio	CustomObject1Id, CustomObject2Id, CustomObject3Id
Portfolio child	CustomObject1Id, CustomObject2Id, CustomObject3Id
Service Request	CustomObject1Id, CustomObject2Id, CustomObject3Id
Solution	CustomObject1Id, CustomObject2Id, CustomObject3Id
Vehicle	CustomObject1Id, CustomObject2Id, CustomObject3Id

### Changes to WSDL Files

To enable the field-level replace operation, the data types for many elements in WSDL files were changed from the string to OperationData data type in Oracle CRM On Demand Release 26.

For example, in the Web services v2.0 WSDL file for Service Request in Release 25:

```
<xsd:element name="CustomObject1Id" maxOccurs="1" minOccurs="0" type="xsd:string" />
```

and for the WSDL for Service Request in Release 26 and later:

```
<xsd:element name="CustomObject1Id" maxOccurs="1" minOccurs="0" type="xsdLocal1:OperationData" />
```

The definition of the OperationData type is as follows:

```
<xsd:complexType name="OperationData">
  <xsd:simpleContent>
```

```

<xsd: extension base="xsd: string">
  <xsd: attribute name="operation" type="xsd: string" />
</xsd: extension>
</xsd: simpleContent>
</xsd: complexType>

```

This change will not regress existing proxy classes and integrations generated using WSDL files from before Release 26. However, if you are planning to generate fresh proxy classes from WSDL files from Release 26 and later, you must enhance your integration to use the `OperationData` data type.

Also, note that upgraded Release 26 proxies and updated integrations will be able to consume the `OperationData` data type value against a Release 25 environment only if the `replace` functionality is not used. In other words, you cannot use the `operation` attribute against an environment based on an older release than Release 26.

## 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
<code>ListOf(Object)</code> . For example, <code>ListOfAccount</code>	The list of object instances to be deleted.	Yes	Not applicable	Input/Output
<code>LOVLanguageMode</code>	The language mode for picklists, see <a href="#">“The LOVLanguageMode Argument” on page 109</a> .	No	LIC	Input



Table 31. Arguments Taken by the Delete 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 <a href="#">“The ViewMode Argument” on page 109</a> .	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 550](#) 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: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:Envelope>
```

```

</soap: Header>
<soap: Body>
  <ContactDelete_Input 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>

```

```

<Modi fi edDate>2011-08-10T15: 21: 43Z</Modi fi edDate>
<CreatedDate>2011-08-09T21: 17: 05Z</CreatedDate>
<Modi fi edByI d>HRZ780-I 5RUE</Modi fi edByI d>
<CreatedByI d>HRZ780-I 5RUE</CreatedByI d>
<ModI d>4</ModI d>
<I d>ADSA-96I ZCS</I d>
<CreatedBy>Joanne Brown, 08/09/2011 16: 17: 05</CreatedBy>
<Modi fi edBy>Joanne Brown, 08/10/2011 10: 21: 43</Modi fi edBy>
<Li stOfAccount>
  <Account>
    <Modi fi edDate>2011-08-10T14: 44: 27Z</Modi fi edDate>
    <CreatedDate>2011-08-10T14: 44: 27Z</CreatedDate>
    <Modi fi edByI d>HRZ780-I 5RUE</Modi fi edByI d>
    <CreatedByI d>HRZ780-I 5RUE</CreatedByI d>
    <ModI d>0</ModI d>
    <I d>ADSA-977GCL</I d>
    <CreatedBy>Joanne Brown, 08/10/2011 09: 44: 27</CreatedBy>
    <Modi fi edBy>Joanne Brown, 08/10/2011 09: 44: 27</Modi fi edBy>
  </Account>
</Li stOfAccount>
</Contact>
<Contact>
  <Modi fi edDate>2011-08-10T15: 22: 46Z</Modi fi edDate>
  <CreatedDate>2011-08-09T21: 13: 33Z</CreatedDate>
  <Modi fi edByI d>HRZ780-I 5RUE</Modi fi edByI d>
  <CreatedByI d>HRZ780-I 5RUE</CreatedByI d>
  <ModI d>2</ModI d>
  <I d>ADSA-96J5ZL</I d>
  <CreatedBy>Joanne Brown, 08/09/2011 16: 13: 33</CreatedBy>
  <Modi fi edBy>Joanne Brown, 08/10/2011 10: 22: 46</Modi fi edBy>

```

```
</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 <a href="#">“The LOVLanguageMode Argument” on page 109.</a>	No	LIC	Input
ViewMode	Specifies the level of access to records specified in the method call, see <a href="#">“The ViewMode Argument” on page 109.</a>	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 550](#) 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: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>
    <wssse:Security>
      <wssse:UsernameToken>
```

```

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

```

## Sample SOAP Response

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

```
<?xml version="1.0" encoding="utf-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>
        <CreateDate>2011-08-25T15:28:34Z</CreateDate>
        <ModifiedById>HRZ780-I5RUE</ModifiedById>
        <CreatedById>HRZ780-I5RUE</CreatedById>
        <ModId>0</ModId>
        <Id>ADSA-9HXB9V</Id>
        <CreatedBy>Joanne Brown, 08/25/2011 10:28:34</CreatedBy>
        <ModifiedBy>Joanne Brown, 08/25/2011 10:28:34</ModifiedBy>
      </Contact>
      <Contact>
        <ModifiedDate>2011-08-25T15:28:36Z</ModifiedDate>
        <CreateDate>2011-08-04T17:01:41Z</CreateDate>
        <ModifiedById>HRZ780-I5RUE</ModifiedById>
        <CreatedById>HRZ780-I5RUE</CreatedById>
        <ModId>5</ModId>
        <Id>ADSA-93DZIC</Id>
        <CreatedBy>Joanne Brown, 08/04/2011 12:01:41</CreatedBy>
        <ModifiedBy>Joanne Brown, 08/25/2011 10:28:36</ModifiedBy>
      </Contact>
      <Contact>
        <ModifiedDate>2011-08-12T15:11:17Z</ModifiedDate>
        <CreateDate>2011-08-12T15:11:17Z</CreateDate>
        <ModifiedById>HRZ780-I5RUE</ModifiedById>
        <CreatedById>HRZ780-I5RUE</CreatedById>
        <ModId>0</ModId>
        <Id>ADSA-98V4V7</Id>
        <CreatedBy>Joanne Brown, 08/12/2011 10:11:17</CreatedBy>
        <ModifiedBy>Joanne Brown, 08/12/2011 10:11:17</ModifiedBy>
      </Contact>
    </ListOfContact>
  </ns:ContactExecute_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        </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.

Insert calls must include values for fields that are required for the object. Fields can be conditionally required, that is they become required if the data specified for other fields on the record meets the criteria defined in an expression for the field. For more information about conditionally-required fields, see *Oracle CRM On Demand Online Help*.

### 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 <a href="#">“The LOVLanguageMode Argument” on page 109</a> .	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 <a href="#">“The ViewMode Argument” on page 109</a> .	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 550](#) 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: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>
    <ContactInsert_Input xmlns="urn:crmondemand/ws/ecbs/contact/">
      <Echo>0n</Echo>
      <LOVLanguageMode>LIC</LOVLanguageMode>
    </ContactInsert_Input>
  </soap:Body>
</soap:Envelope>
```

```

<ListOfContact>
  <Contact>
    <ContactFirstName>NewCon2</ContactFirstName>
    <ContactLastName>Con2</ContactLastName>
    <ExternalSystemId>12345009</ExternalSystemId>
  </Contact>
  <Contact>
    <ContactFirstName>Exi sti ngFi rstName</ContactFirstName>
    <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>
          <CreatedDate>2011-08-10T14:44:24Z</CreatedDate>
          <ModifiedBy>HRZ780-I5RUE</ModifiedBy>
          <CreatedBy>HRZ780-I5RUE</CreatedBy>
          <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>
  <Modi fi edDate>2011-08-10T14: 41: 55Z</Modi fi edDate>
  <CreatedDate>2011-08-09T21: 17: 05Z</CreatedDate>
  <Modi fi edByI d>HRZ780-I 5RUE</Modi fi edByI d>
  <CreatedByI d>HRZ780-I 5RUE</CreatedByI d>
  <ModI d>2</ModI d>
  <I d>ADSA-96I ZCS</I d>
  <CreatedBy>Joanne Brown, 08/09/2011 16: 17: 05</CreatedBy>
  <Modi fi edBy>Joanne Brown, 08/10/2011 09: 41: 55</Modi fi edBy>
  <Li stOfAccount>
    <Account>
      <Modi fi edDate>2011-08-10T14: 44: 27Z</Modi fi edDate>
      <CreatedDate>2011-08-10T14: 44: 27Z</CreatedDate>
      <Modi fi edByI d>HRZ780-I 5RUE</Modi fi edByI d>
      <CreatedByI d>HRZ780-I 5RUE</CreatedByI d>
      <ModI d>0</ModI d>
      <I d>ADSA-977GCL</I d>
      <CreatedBy>Joanne Brown, 08/10/2011 09: 44: 27</CreatedBy>
      <Modi fi edBy>Joanne Brown, 08/10/2011 09: 44: 27</Modi fi edBy>
    </Account>
  </Li stOfAccount>
</Contact>
</Li stOfContact>
</ns: ContactI nsert_Output>
</SOAP-ENV: Body>
</SOAP-ENV: Envel ope>

```

## 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.

### Team and Role Visibility Considerations

If a query includes a child object for which the user does not have access due to team or role visibility, then the child object returns no records.

### 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 131](#).

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.

**TIP:** It is recommended to use a stateful rather than a stateless `QueryPage` Web service request, if it is required to paginate through a snapshot of data. However, if you do use a stateless request, it is recommended to use sort criteria to reduce the possibility of returning the same record when paging through results using the `QueryPage` method. For more information, see [“Use Sort Criteria when Using QueryPage” on page 70](#).

## 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
<b>expressi on</b>	
[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.
<b>conj uncti on</b>	
OR	None
AND	None
<b>unary operator</b>	
IS NULL	Used to find a match for a value that has no value
IS NOT NULL	None
EXISTS	See <a href="#">“The EXISTS and NOT EXISTS Operators” on page 130</a>
NOT EXISTS	None
<b>bi nary operator</b>	
=	None
~=	Denotes a case-insensitive exact search (no wildcards used)
<	Must be specified as &lt; to ensure well-formed XML.
<=	Must be specified as &lt; =
>	None
>=	None
<>	Must be specified as &lt; >
LI KE	Wildcard characters are treated as such only in the context of the operator LI KE.
-LI KE	Denotes a case-insensitive wildcard search

Table 34. Query Syntax for QueryPage

Syntax Type	Notes
value	
' <i>literal</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. In some cases, you can however use shadow fields, see ["Using Shadow Fields for Better Search Performance"](#) on page 130.

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>&lt;ListOfContact startrownum="0" pagesize="100" recordcountneeded="true"&gt; &lt;Contact searchspec="[ContactFirstName] LIKE 'Contact*' "&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with zero or more characters
Case Insensitive ~ LIKE with * wildcard	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec="[ContactFirstName] ~LIKE 'Contact*' "&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</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>&lt;ListOfContact&gt;&lt;Contact searchspec="[ContactFirstName] LIKE 'Contact???' "&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with any three characters
Case Insensitive ~ LIKE with ? wildcard	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec="[ContactFirstName] ~LIKE 'Contact???' "&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</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>&lt;ListOfContact&gt;&lt;Contact searchspec="[ContactFirstName] = 'ContactInsert' "&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose ContactFirstName value is equal to "ContactInsert"
Case Insensitive ~=	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec="[ContactFirstName] ~= 'ContactInsert' "&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose ContactFirstName value is equal to, for example, "ContactInsert", "contactinsert" or "CoNtAcTiNsErT"
IS NULL	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec="[TEXTLG_000] IS NULL"&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</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>&lt;ListOfContact&gt;&lt;Contact searchspec=" [TEXTLG_000] IS NOT NULL"&gt; &lt;ContactFirstName/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose TEXTLG_000 value is not NULL
AND	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec=" [BOOL_000] = ' Y' AND [IndexedBoolean0] = ' N' "&gt; &lt;ContactFirstName/&gt;&lt;IndexedBoolean0/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre> <p>Or</p> <pre>&lt;ListOfContact&gt; &lt;Contact searchspec=" [BOOL_000] = ' Y' "&gt; &lt;ContactFirstName/&gt; &lt;IndexedBoolean0&gt;=' N' &lt;/IndexedBoolean0&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre> <p>Or</p> <pre>&lt;ListOfContact&gt;&lt;Contact&gt; &lt;ContactFirstName/&gt; &lt;BOOL_000&gt;=' Y' &lt;/BOOL_000&gt; &lt;IndexedBoolean0&gt;=' N' &lt;/IndexedBoolean0&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose BOOL_000 value is Y AND IndexedBoolean0 value is N
OR	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec=" [BOOL_000] = ' Y' OR [IndexedBoolean0] = ' N' "&gt; &lt;ContactFirstName/&gt; &lt;BOOL_000/&gt; &lt;IndexedBoolean0/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose BOOL_000 value is Y OR IndexedBoolean0 value is N
> (greater than)	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec=" [IndexedNumber0] &gt; ' 500' "&gt; &lt;ContactFirstName/&gt; &lt;IndexedNumber0/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</pre>	Returns all contacts whose IndexedNumber0 value is greater than 500
>= (greater than or equal to)	<pre>&lt;ListOfContact&gt;&lt;Contact searchspec=" [IndexedNumber0] &gt;= ' 500' "&gt; &lt;ContactFirstName/&gt; &lt;IndexedNumber0/&gt; &lt;/Contact&gt;&lt;/ListOfContact&gt;</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)	<ListOfContact><Contact searchspec="[IndexedNumber0] < '500' "> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact>	Returns all contacts whose IndexedNumber0 value is less than 500
<= (less than or equal to)	<ListOfContact><Contact searchspec="[IndexedNumber0] <= '500' "> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact>	Returns all contacts whose IndexedNumber0 value is less than or equal to 500
<> (not equal to)	<ListOfContact><Contact searchspec="[IndexedNumber0] <> '500' "> <ContactFirstName/> <IndexedNumber0/> </Contact></ListOfContact>	Returns all contacts whose IndexedNumber0 value is not equal to 500

### Using Shadow Fields for Better Search Performance

To provide better performance for case-insensitive wildcard searches, you can use shadow fields in search specifications for some record types. Shadow fields are predefined and are found in the WSDL file for each object for which they are supported.

Shadow fields add a suffix, `_Shadow`, to the actual field name. For example, for a `Name` field there would be a `Name_Shadow` field. Shadow fields always store their value in upper case, so, for example, if the `Name` field has the value `Oracle`, then the `Name_Shadow` field has the value `ORACLE`.

To perform case insensitive searches, you use the shadow field instead of the actual field. For example, instead of a searchspec like this:

```
searchspec="[Name] ~LIKE '%cl e'"
```

you can use the following searchspec including upper case characters for better performance:

```
searchspec="[Name_Shadow] LIKE '%CLE'"
```

### 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>&lt;ListOfContact startrownum="0" page size="100" recordcountneeded="true"&gt; &lt;Contact searchspec="[ContactFi rstName] LIKE ' Contact*' "&gt; &lt;ContactLastName sortorder="ASC"&gt;&lt;/ ContactLastName&gt; &lt;/Contact&gt;&lt;/Li stOfContact&gt;</pre>	Returns all contacts whose ContactFirstName value starts with "Contact" and ends with zero or more characters and sorts them by ContactLastName in ascending order
Sort in descending order	<pre>&lt;ListOfContact startrownum="0" page size="100" recordcountneeded="true"&gt; &lt;Contact&gt; &lt;ContactLastName sortorder="DESC"&gt;-LIKE ' Contact*' &lt;/ContactLastName&gt; &lt;/Contact&gt;&lt;/Li stOfContact&gt;</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>&lt;ListOfContact startrownum="0" page size="100" recordcountneeded="true"&gt; &lt;Contact searchspec="[ContactFi rstName] LIKE ' Contact*' "&gt; &lt;ContactLastName sortorder="DESC" sortsequence="1"&gt;&lt;/ContactLastName&gt; &lt;ContactFi rstName sortorder="DESC" sortsequence="2"&gt;&lt;/ContactFi rstName&gt; &lt;ExternalSystemI d sortorder="DESC" sortsequence="3"&gt;&lt;/ExternalSystemI d&gt; &lt;/Contact&gt;&lt;/Li stOfContact&gt;</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 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.

You can use `NamedSearchSpec` together with the `searchspec` argument to search within a list. In a `QueryPage` request when `NamedSearchSpec` and field-level `searchspecs` are specified, an AND operation is performed between `NamedSearchSpec` and the field-level `searchspecs`. For example, if the `searchspec` in a request is:

```
"[Field]=500"
```

and the value of `NamedSearchSpec` is:

```
(([Field1] > 100 OR ([Field2] > 500 AND [Field 3] = true) AND [Field4] = 'MyName') AND [Field5] = 'Open')
```

then the effective `searchspec` is:

```
(([Field]=500) AND (([Field1] > 100 OR ([Field2] > 500 AND [Field 3] = true) AND [Field4] = 'MyName') AND [Field5] = 'Open'))
```

## Specifying Proximity Search Parameters

In the Maps page in Oracle CRM On Demand you can perform a proximity search to check the addresses on records of a given record type to determine whether the addresses are within a specified radius of a selected location. Proximity searches are supported for the Account, Activity, and Contact record types. In the `QueryPage` methods for these record types, you must use the following arguments to specify the parameters for a proximity search:

- **Latitude.** The latitude value of the proximity search center.
- **Longitude.** The longitude value of the proximity search center.
- **Radius.** The radius of the proximity search. Possible values are: 1,5,10,20,25.
- **Units.** The unit of distance: Miles or Kilometers.
- **AddressType.** For the Account record type only, the address type for the proximity search:
  - **BillingAddress.** The proximity search for the account is limited to the billing address.
  - **ShippingAddress.** The proximity search for the account is limited to the shipping address.

If the `AddressType` parameter is not provided or is empty, the proximity search is limited to the shipping address.

The `namedsearchspec` argument is used to define lists of records on which proximity searches can be made. If the arguments are missing or incorrect, an error message is received.

For more information about proximity searching, see *Oracle CRM On Demand Online Help*.

## 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 pagesize=100 and 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.  <b>NOTE:</b> 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 <a href="#">“Using the pagesize, startrownum, and recordcountneeded Arguments” on page 124.</a>	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 <a href="#">“The LOVLanguageMode Argument” on page 109.</a>	No	LIC	Input
ViewMode	Specifies the level of access to records specified in the method call, see <a href="#">“The ViewMode Argument” on page 109.</a>	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, with attributes as follows:

- **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: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>
  <body>
    <QueryPage>
      <pagesize>10</pagesize>
      <startrownum>1</startrownum>
      <recordcountneeded>true</recordcountneeded>
      <ViewMode>BROADEST</ViewMode>
      <LOVLanguageMode>LIC</LOVLanguageMode>
    </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>
  <ContactEmai l ></ContactEmai l >
  <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>
  <ContactEmai l ></ContactEmai l >
  <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>
  <ContactEmai l >n@l . com</ContactEmai l >
  <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 ></ContactEmail >
    <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: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>
    <wssse:Security>
      <wssse:UsernameToken>
        <wssse:Username>%%USERNAME%%</wssse:Username>
        <wssse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wssse:Password>
      </wssse:UsernameToken>
    </wssse: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: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="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>Mari anne</ContactFirstName>
  <ContactLastName>Pok</ContactLastName>
  <ContactType>Customer</ContactType>
</Contact>
<Contact>
  <ContactFirstName>Bri an</ContactFirstName>
  <ContactLastName>Pi ttenger</ContactLastName>
  <ContactType>Customer</ContactType>
</Contact>
<Contact>
  <ContactFirstName>Mari e</ContactFirstName>
  <ContactLastName>Perki ns</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

Table 38. Example of Records in a Contact List

Row	Last Name	First Name	Contact Type	Owner Alias
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>Connors</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>
```



```
</SOAP-ENV: Body>
</SOAP-ENV: Envelope>
```

## Update

Updates the selected record with the new values.

### 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 <a href="#">“The LOVLanguageMode Argument” on page 109</a> .	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 <a href="#">“The ViewMode Argument” on page 109</a> .	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 550](#) 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>0n</Echo>
      <LOVLanguageMode>LIC</LOVLanguageMode>
      <ListOfContact>
        <Contact>
          <ContactFirstName>ExistingCon1</ContactFirstName>
          <ContactLastName>ExistingCon1</ContactLastName>
          <ContactEmail>e@e.com</ContactEmail>
        </Contact>
      </ListOfContact>
    </ContactUpdate_Input>
  </soap:Body>
</soap:Envelope>
```

## Sample SOAP Response

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

```
<?xml version="1.0" encoding="utf-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>
          <ModifiedById>HRZ780-I5RUE</ModifiedById>
          <CreatedById>HRZ780-I5RUE</CreatedById>
          <ModId>8</ModId>
          <Id>ADSA-96I ZCS</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
<a href="#">"AuditTrailQueryPage" on page 149</a>	Audit Trail	Gets records from the master audit trail.
<a href="#">"AuthAuthorizationUser" on page 155</a>	Authorization	Authenticates the current user and returns information about the user's company and the privileges enabled for the user's role.

Table 40. Web Services On Demand Service Methods

Method Name	Web Service	Comments
<a href="#">“DeletedItemQueryPage” on page 157</a>	Deleted Item	Gets information about deleted items.
<a href="#">“DeleteEvents” on page 163</a>	Integration Event	Deletes events from an integration event queue.
<a href="#">“ExportWebServiceCreateExportRequest” on page 166</a>	Export	Creates an export request.
<a href="#">“ExportWebServiceMonitorExportRequest” on page 169</a>	Export	Gets data about an export request.
<a href="#">“ExportWebServiceRetrieveExportAttachment” on page 171</a>	Export	Gets an attachment for an export request.
<a href="#">“GetEvents” on page 173</a>	Integration Event	Gets events from an integration event queue.
<a href="#">“GetLists” on page 182</a>	Lists	Gets the set of lists for an object.
<a href="#">“GetMapping” on page 185</a>	Mapping Service	Gets a list of the display names of fields for a particular record type and their associated XML tags.
<a href="#">“GetPicklistValues” on page 189</a>	Picklist	Gets lists of picklist values.
<a href="#">“GetServerTime” on page 192</a>	Time	Gets the server time.
<a href="#">“LeadConvert” on page 194</a>	Lead Conversion	Converts leads.
<a href="#">“LoginHistoryQueryPage” on page 200</a>	Login History	Gets information about user login history.
<a href="#">“MergeRecords” on page 203</a>	Merge	Merges records.
<a href="#">“MetadataChangeSummaryQueryPage” on page 207</a>	Metadata Change Summary	Gets a summary of changes to metadata.
<a href="#">“SalesProcessQueryPage” on page 210</a>	Sales Process	Gets sales process information.
<a href="#">“SetPassword” on page 216</a>	Password	Sets the passwords of users who use Oracle CRM On Demand.
<a href="#">“SetSessionTimeZone” on page 218</a>	Time	Sets the time zone for a session.
<a href="#">“UpdatePicklist” on page 220</a>	Picklist	Updates picklist values.
<a href="#">“UpdateCascadingPicklists” on page 224</a>	Picklist	Updates cascading picklist values.
<a href="#">“UserUsageQueryPage” on page 229</a>	User Usage	Gets information about Web services utilization.

## AuditTrailQueryPage

Returns records from the master audit trail.

### Usage

You use the AuditTrailQueryPage method to execute a query against the master audit trail, and return a subset of the records that match the search criteria set by the method arguments.

The following are attributes of the <ListOfAuditTrail> element that you can specify:

- **pagesize.** Determines the maximum number of records displayed on a page following a query.
- **startrownum.** Indicates the row from which the AuditTrailQueryPage method starts to return records.
- **recordcountneeded.** Indicates whether a record count for the query is to be returned.

You can also use the searchspec attribute on the AuditTrail element to provide a search specification.

Each of the child elements of <ListOfAuditTrail> can have the following attributes:

- **sortorder.** Determines the sort order for the records returned by the query, either ASC for ascending or DESC for descending.
- **sortsequence.** An integer value that determines the order of the sort specification.

The following are attributes of the <AuditTrailData> element that can be returned:

- **recordcount.** An integer value that indicates the record count.
- **lastpage.** A value that indicates whether or not the last value in the query set has been returned.

### Arguments

Table 41 describes the arguments taken by the AuditTrailQueryPage method.

Table 41. Arguments Taken by the AuditTrailQueryPage Method

Name	Description	Required	Default	I/O
ListOfAuditTrail	The list of audit trail entries queried (input), and after query execution, the list of audit trail entries returned (output).	Yes	Not applicable	Input/Output
NamedSearchSpec	Indicates the search specification for a named list for the audit trail.	No	Not applicable	Input

Table 41. Arguments Taken by the AuditTrailQueryPage Method

Name	Description	Required	Default	I/O
LOVLanguageMode	The language mode for the audit trail, see <a href="#">“The LOVLanguageMode Argument” on page 109.</a>	No	LIC	Input
ViewMode	Specifies the level of access to records specified in the method call, see <a href="#">“The ViewMode Argument” on page 109.</a>	Yes	Broadest	Input

### Return Value of the Call

The following information is returned:

- **Id.** The table row ID.
- **SourceType.** A value that indicates whether the session was interactive or a Web services session.
- **RecordType.** The record type.
- **RecordUpdatedDate.** The date when the object was last updated.
- **UserFirstName.** The user’s first name.
- **UserFullName.** The user’s full name.
- **UserID.** The user’s ID.
- **UserLastName.** The user’s last name.
- **UserSignInID.** The user’s login ID.
- **FieldName.** The field modified in the object.
- **RecordName.** The name of the record.
- **Operation.** The type of change that was performed, for example, create, delete, or modify.
- **SourceIPAddress.** The IP address of the computer on which the change took place.  
**NOTE:** This value is returned only when this information is available.
- **NewFieldValue.** The new value of the field.
- **OldFieldValue.** The previous value of the field.

There are also various elements with names beginning with Created, Modified, and Updated that give information about who performed the create, modify, or update operation on the record.

### Sample SOAP Request - AuditTrailQueryPage

The following SOAP request queries the audit trail. It demonstrates the use of the pagesize and startrownum attributes to return a specific number of records starting at a specific row. The searchspec attribute specifies that only records modified after a certain date and time are to be returned:

```

<?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>
    <wssse:Security>
      <wssse:UsernameToken>
        <wssse:Username>%%USERNAME%%</wssse:Username>
        <wssse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wssse:Password>
      </wssse:UsernameToken>
    </wssse:Security>
  </soap:Header>
  <soap:Body>
    <AuditTrailQueryPage_Input xmlns="urn:crmondemand/ws/audittrail/">
      <ListOfAuditTrail startrownum="0" pagesize="100" recordcountneeded="true">
        <AuditTrail searchspec="[ModifiedDate] > '2013-06-02T07:52:59Z' ">
          <ModifiedDate/>
          <CreateDate/>
          <ModifiedById/>
          <CreatedById/>
          <ModId/>
          <Id/>
          <SourceType/>
          <RecordType/>
          <RecordUpdatedDate/>
          <UserFirstName/>
          <UserFullName/>
          <UserId/>
          <UserLastName/>
          <UserSigninId/>
          <FieldName/>
          <RecordName/>
          <Operation/>
          <SourceIPAddress/>
          <UpdatedByFirstName/>
          <UpdatedByLastName/>
          <UpdatedByUserSigninId/>
        </AuditTrail>
      </ListOfAuditTrail>
    </AuditTrailQueryPage_Input>
  </soap:Body>
</soap:Envelope>

```

```

    <UpdatedByAlias/>
    <UpdatedByFullName/>
    <UpdatedByIntegrationId/>
    <UpdatedByExternalSystemId/>
    <UpdatedByEmailAddr/>
    <CreatedByFirstName/>
    <CreatedByLastName/>
    <CreatedByUserSignatureId/>
    <CreatedByAlias/>
    <CreatedByFullName/>
    <CreatedByIntegrationId/>
    <CreatedByExternalSystemId/>
    <CreatedByEmailAddr/>
    <CreatedBy/>
    <ModifiedBy/>
    <NewFieldValue/>
    <OldFieldValue/>
  </AuditTrail>
</ListOfAuditTrail>
</AuditTrailQueryPage_Input>
</soap:Body>
</soap:Envelope>

```

## Sample SOAP Response

The following illustrates the SOAP response (most of the <AuditTrail> 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:AuditTrailQueryPage_Output>
      <ListOfAuditTrail recordcount="24" lastpage="true">
        <AuditTrail>
          <ModifiedDate>2014-06-02T07:52:59Z</ModifiedDate>
          <CreateDate>2014-06-02T07:52:59Z</CreateDate>
          <ModifiedById>1-D5S9Z</ModifiedById>
          <CreatedById>1-D5S9Z</CreatedById>

```



```

<ModId>0</ModId>
<Id>1QA2-22RN1C</Id>
<SourceType>UI </SourceType>
<RecordType>Opportunity</RecordType>
<RecordUpdatedDate>2014-06-02T07: 52: 58Z</RecordUpdatedDate>
<UserFirstName>System</UserFirstName>
<UserFullName>SystemAdmin</UserFullName>
<UserId>1-D5S9Z</UserId>
<UserLastName>Admin</UserLastName>
<UserSgnlId>ADMIN@CUSTCOMP2.COM</UserSgnlId>
<FieldName>Sales Stage - Translation</FieldName>
<RecordName>Testing</RecordName>
<Operation>Modify</Operation>
<SourceIPAddress>12. 168. 23. 232</SourceIPAddress>
<UpdatedByFirstName>System</UpdatedByFirstName>
<UpdatedByLastName>Admin</UpdatedByLastName>
<UpdatedByUserSgnlId>ADMIN@CUSTCOMP2.COM</UpdatedByUserSgnlId>
<UpdatedByAlias>System Admin Alias (admin@custcomp2.com)</UpdatedByAlias>
<UpdatedByFullName>System Admin</UpdatedByFullName>
<UpdatedByIntegratId>1-D5S9W</UpdatedByIntegratId>
<UpdatedByExternalSystemId/>
<UpdatedByEmailAddr>wiley.coyote@acme.com</UpdatedByEmailAddr>
<CreatedByFirstName>System</CreatedByFirstName>
<CreatedByLastName>Admin</CreatedByLastName>
<CreatedByUserSgnlId>ADMIN@CUSTCOMP2.COM</CreatedByUserSgnlId>
<CreatedByAlias>System Admin Alias (admin@custcomp2.com)</CreatedByAlias>
<CreatedByFullName>System Admin</CreatedByFullName>
<CreatedByIntegratId>1-D5S9W</CreatedByIntegratId>
<CreatedByExternalSystemId/>
<CreatedByEmailAddr>wiley.coyote@acme.com</CreatedByEmailAddr>
<CreatedBy>System Admin, 06/02/2014 07: 52: 59</CreatedBy>
<ModifiedBy>System Admin, 06/02/2014 07: 52: 59</ModifiedBy>
<NewFieldVal ue>Short List</NewFieldVal ue>
<OldFieldVal ue>Qualified Lead</OldFieldVal ue>
</AuditTrail >

```

...

...

```
<AuditTrail>
  <ModifiedDate>2014-06-02T08:06:10Z</ModifiedDate>
  <CreateDate>2014-06-02T08:06:10Z</CreateDate>
  <ModifiedBy>1-D5S9Z</ModifiedBy>
  <CreatedBy>1-D5S9Z</CreatedBy>
  <ModId>0</ModId>
  <Id>10A2-22RN1L</Id>
  <SourceType>UI</SourceType>
  <RecordType>x</RecordType>
  <RecordUpdatedDate>2014-06-02T08:06:10Z</RecordUpdatedDate>
  <UserFirstName>System</UserFirstName>
  <UserFullName>System Admin</UserFullName>
  <UserId>1-D5S9Z</UserId>
  <UserLastName>Admin</UserLastName>
  <UserSignInId>ADMIN@CUSTCOMP2.COM</UserSignInId>
  <FieldName>Account/AuditTrailItem2</FieldName>
  <RecordName>Administrator Owner Access Profile</RecordName>
  <Operation>Modify</Operation>
  <SourceIPAddress>12.168.23.232</SourceIPAddress>
  <UpdatedByFirstName>System</UpdatedByFirstName>
  <UpdatedByLastName>Admin</UpdatedByLastName>
  <UpdatedByUserSignInId>ADMIN@CUSTCOMP2.COM</UpdatedByUserSignInId>
  <UpdatedByAlias>System Admin Alias (admin@custcomp2.com)</UpdatedByAlias>
  <UpdatedByFullName>System Admin</UpdatedByFullName>
  <UpdatedByIntegratorId>1-D5S9W</UpdatedByIntegratorId>
  <UpdatedByExternalSystemId/>
  <UpdatedByEmailAddr>wiley.coyote@acme.com</UpdatedByEmailAddr>
  <CreatedByFirstName>System</CreatedByFirstName>
  <CreatedByLastName>Admin</CreatedByLastName>
  <CreatedByUserSignInId>ADMIN@CUSTCOMP2.COM</CreatedByUserSignInId>
  <CreatedByAlias>System Admin Alias (admin@custcomp2.com)</CreatedByAlias>
  <CreatedByFullName>System Admin</CreatedByFullName>
  <CreatedByIntegratorId>1-D5S9W</CreatedByIntegratorId>
  <CreatedByExternalSystemId/>
  <CreatedByEmailAddr>wiley.coyote@acme.com</CreatedByEmailAddr>
```

```

        <CreatedBy>System Admin, 06/02/2014 08:06:10</CreatedBy>
        <ModifiedBy>System Admin, 06/02/2014 08:06:10</ModifiedBy>
        <NewFieldValue>1</NewFieldValue>
        <OldFieldValue>-1</OldFieldValue>
    </AuditTrail>
</ListOfAuditTrail>
</ns:AuditTrailQueryPage_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
    
```

## 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 42 describes the arguments taken by the AuthAuthorizationUser method.

Table 42. 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: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>
    <AuthWS_AuthAuthorizationUser_Input xmlns="urn:crmondemand/ws/auth/" />
  </soap:Body>
</soap:Envelope>
```

### Sample SOAP Response

The following illustrates the SOAP response (most of the <Privilege> elements are omitted as indicated by the ellipses (...)):

```
<?xml version="1.0" encoding="utf-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 set of records returned is sorted by DeletedDate in ascending order (as shown in the response in ["Sample SOAP Request - DeletedItemQueryPage" on page 161](#)).

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 43.

**NOTE:** In Table 43, 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 43. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
Account	Account
Account Attachment	Account Attachment
Account Note	Account Note
Account Private Note	Account Private Note
Action Attachment	Action Attachment
Address	Address
Appointment	Action***Appointment
Asset	Asset Mgmt - Asset
Campaign	Campaign
Campaign Attachment	Campaign Attachment
Campaign Note	Campaign Note
Channel Partner	Channel Partner
Contact	Contact
Contact Attachment	Contact Attachment
Contact Interest	Contact Interests
Contact Note	Contact Note
Contact Private Note	Contact Private Note
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)
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
Custom Object 01 Note	OnDemand Custom Object 1 Note

Table 43. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
Custom Object 02 Note	OnDemand Custom Object 2 Note
Custom Object 03 Note	OnDemand Custom Object 3 Note
Dealer Attachment	Dealer Attachment
Dealer Note	Dealer Note
Event	Pharma ME Event
Forecast Revenue	Forecast Revenue
Fund	Fund
Fund Attachment	Fund Attachment
Fund Note	Fund Note
Fund Request	Fund Request
Fund Request Attachment	Fund Request Attachment
Fund Request Note	Fund Request Note
Household	Household
Lead	Lead
Lead Attachment	Lead Attachment
Note	Note
Opportunity	Opportunity
Opportunity Attachment	Opportunity Attachment
Opportunity Note	Opportunity Note
Opportunity Private Note	Opportunity Private Note
Organization	Organization
Organizations Note	Organizations Note
Plan Account	CRMODLS_BPL_ACNT
Portfolio	VONDINS Portfolio***Portfolio
Portfolio Child	VONDINS Portfolio Child***Portfolio
Position	Position
Product Detailed	Pharma Call Product Detailed
Referral	VONDINS Referral***Referral
Revenue	Revenue
Sales Stage Attachment	Sales Stage Attachment
Sample Dropped	Pharma Call Sample Dropped

Table 43. Deleted Item Types Returned by DeletedItemQueryPage Method

UI Record Type	Deleted Item Type
Service Request	Service Request
Service Request Attachment	Service Request Attachment
Service Request Note	Service Request Note
Smart Call	Pharma Template Call
Solution	Solution
Solution Attachment	Solution Attachment
Task	Action***Task
Vehicle	Auto Vehicle

## Arguments

Table 44 describes the arguments taken by the DeletedItemQueryPage method.

Table 44. Arguments Taken by the DeletedItemQueryPage Method

Name	Description	Required	Default	I/O
ListOfDeletedItem	The list of object instances queried (input), and after query execution, the list of object instances returned (output).	Yes	Not applicable	Input/Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the DeletedItemQueryPage method starts to return records. Use the StartRowNum argument to return a set of records for any given method.  For example, if PageSize=100 and 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:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%USERNAME%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <DeletedItemWS_DeletedItemQueryPage_Input xmlns="urn:crmondemand/ws/deleteditem/">
      <ListOfDeletedItem>
        <DeletedItem>
          <DeletedItemId />
          <DeletedById />
          <DeletedBy />
          <DeletedDate />
          <Name />
        </DeletedItem>
      </ListOfDeletedItem>
    </DeletedItemWS_DeletedItemQueryPage_Input>
  </soap:Body>
</soap:Envelope>
```

```

        <Obj ectI d />
        <Type>=' Contact' </Type>
        <External SystemI d />
    </Del etedI tem>
</Li stOfDel etedI tem>
<PageSi ze>3</PageSi ze>
<StartRowNum>0</StartRowNum>
</Del etedI temWS_Del etedI temQueryPage_I nput>
</soap: Body>
</soap: Envel ope>

```

### Sample SOAP Response

The following SOAP response contains three deleted contact records:

```

<?xml versi on="1. 0" encodi ng="utf-8"?>
<SOAP-ENV: Envel ope xml ns: SOAP-ENV="http: //schemas. xml soap. org/soap/envel ope/" xml ns: xsi ="http: //
www. w3. org/2001/XMLSchema-i nstance" xml ns: xsd="http: //www. w3. org/2001/XMLSchema">
  <SOAP-ENV: Body>
    <ns: Del etedI temWS_Del etedI temQueryPage_Output xml ns: ns="urn: crmondemand/ws/del etedi tem/">
      <ns: LastPage>true</ns: LastPage>
      <Li stOfDel etedI tem xml ns="urn: /crmondemand/xml /del etedi tem">
        <Del etedI tem>
          <Del etedI temI d>ADSA-921PLC</Del etedI temI d>
          <Del etedBy>Joanne Brown</Del etedBy>
          <Del etedByI d>HRZ780-I 5RUE</Del etedByI d>
          <Del etedDate>08/02/2011 15: 19: 24</Del etedDate>
          <Name>Person One</Name>
          <Obj ectI d>ADSA-921PL5</Obj ectI d>
          <Type>Contact</Type>
          <External SystemI d></External SystemI d>
        </Del etedI tem>
        <Del etedI tem>
          <Del etedI temI d>ADSA-93EVM</Del etedI temI d>
          <Del etedBy>Joanne Brown</Del etedBy>
          <Del etedByI d>HRZ780-I 5RUE</Del etedByI d>
          <Del etedDate>08/04/2011 12: 16: 39</Del etedDate>
          <Name>Jordan Lak</Name>
          <Obj ectI d>ADSA-93EGXJ</Obj ectI d>

```

```

    <Type>Contact</Type>
    <External SystemId>ABC321</External SystemId>
  </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-93ET51</ObjectId>
    <Type>Contact</Type>
    <External SystemId>abc223</External SystemId>
  </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 all objects that use Web Services v2.0 Schema and the following objects that use Web Services v1.0 Schema:

Account, Account Competitor, Account Contact, Account Contact Role, Account Note, Account Team, Activity, Activity Attachment, Asset, Call PromItemDrop, Call SampleDropped, Campaign, Contact, Contact Campaign, Contact Interests, Contact Note, Contact Team, Custom Object 01 - 03, Custom Object 01 - 03 Team, Dealer, Event Household, Household Contact, HouseholdTeam, Lead, Lead Team, Opportunity, Opportunity Competitor, Opportunity Contact Role, Opportunity Note, Opportunity Partner, Opportunity Team, Portfolio, Portfolio Owner, PortfolioTeam, Product, ProductCategory, Program, Program Partner, Service Request, Service Request Note, Service Request Team, Solution, Territory, 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 181](#).

## Arguments

[Table 45](#) describes the arguments taken by the DeleteEvents method.

Table 45. 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>
```

## 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 46 describes the arguments taken by the ExportWebServiceCreateExportRequest method.

Table 46. 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
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
EnableMailNotificationflag	Whether email notifications are enabled (true) or disabled (false).	No	Not applicable	Input
IncludeWebLinkURL	Whether URLs are included for Weblinks in the export request (true) or excluded (false).	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: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>
    <ExportWebServiceCreateExportRequest_Input xmlns="urn:crmondemand/ws/exportwebsevice">
      <TimeZone>(GMT-08:00) Pacific Time (US & Canada); Tijuana</TimeZone>
      <ListOfRecordType>
        <RecordType>
          <Name>Contact</Name>
        </RecordType>
        <RecordType>
          <Name>Account</Name>
        </RecordType>
      </ListOfRecordType>
      <EnableMailNotifcationFlag>false</EnableMailNotifcationFlag>
      <IncludeWebLinkURL>false</IncludeWebLinkURL>
      <DateTimeFormat>USA - MM/DD/YYYY hh:mm:ss PM</DateTimeFormat>
    </ExportWebServiceCreateExportRequest_Input>
  </soap:Body>
</soap:Envelope>
```

```

    <Del i mi ter></Del i mi ter>

    </ExportWebServi ce_ExportWebServi ceCreateExportRequest_I nput>
</soap: Body>
</soap: Envel ope>

```

### 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 versi on="1.0" encodi ng="utf-8" standal one="no"?>
<soap: Envel ope xml ns: soap="http://schemas.xml soap.org/soap/envel ope/" xml ns: xsi ="http://www.w3.org/2001/
XMLSchema-i nstance" xml ns: wsse="http://docs.oasi s-open.org/wss/2004/01/oasi s-200401-wss-wssecuri ty-
secext-1.0.xsd" xml ns: xsd="http://www.w3.org/2001/XMLSchema">
  <soap: Header>
    <wsse: Securi ty>
      <wsse: UsernameToken>
        <wsse: Username>%%USERNAME%%</wsse: Username>
        <wsse: Password Type="http://docs.oasi s-open.org/wss/2004/01/oasi s-200401-wss-username-token-
profi le-1.0#PasswordText">%%PASSWORD%%</wsse: Password>
      </wsse: UsernameToken>
    </wsse: Securi ty>
  </soap: Header>
  <soap: Body>
    <ExportWebServi ce_ExportWebServi ceCreateExportRequest_I nput xml ns="urn: crmondemand/ws/
exportwebservi ce/">
      <ExportDataModi fi edBefore>12/07/2011</ExportDataModi fi edBefore>
      <Ti meZone>(GMT-08:00) Paci fi c Ti me (US & amp; Canada); Ti juana</Ti meZone>
      <Li stOfRecordType>
        <RecordType>
          <Name>Contact</Name>
        </RecordType>
      </Li stOfRecordType>
      <Enabl eMai l Noti fi cati onfl ag>fal se</Enabl eMai l Noti fi cati onfl ag>
      <I ncl udeWebLi nkURL>fal se</I ncl udeWebLi nkURL>
      <DateTi meFormat>USA - MM/DD/YYYY hh:mm:ss PM</DateTi meFormat>
      <ExportDataModi fi edAfter>12/06/2011</ExportDataModi fi edAfter>
      <Del i mi ter>Semi -Col on</Del i mi ter>
    </ExportWebServi ce_ExportWebServi ceCreateExportRequest_I nput>

```



```
</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 47 describes the arguments taken by the ExportWebServiceMonitorExportRequest method.

Table 47. 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>%%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_ExportWebServiceMonitorExportRequest_Input xmlns="urn:crmondemand/ws/exportwebservice/">
        <RequestId>10A2-16438E</RequestId>
        </ExportWebService_ExportWebServiceMonitorExportRequest_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:ExportWebService_ExportWebServiceMonitorExportRequest_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>10A2-166416</FileId>
                    <FileName>OnDemandExport_20111207_10A2-16438E_0001</FileName>
                    <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 48 describes the arguments taken by the `ExportWebServiceRetrieveExportAttachment` method.

Table 48. 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: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>
  <body>
    <ExportWebServiceRetrieveExportAttachment RequestId="1" AttachmentId="1" />
  </body>
</soap:Envelope>
```

```

    </wsse: Security>
  </soap: Header>
  <soap: Body>
    <ExportWebService_ExportWebServiceRetrieveExportAttachment_Input xmlns="urn:crmondemand/ws/exportwebservice/">
      <RequestId>10A2-16438E</RequestId>
      <AttachmentId>10A2-166416</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>UESDBBQAAAAIAAhOhz/
CCmOj UAEA AHKCAA mAHEARXhwb3JOU3VtbWFyeV8yMDExMTIwN18xUUeYlTE2NDM4RS50eHRTRFwApAAAAAAIAA6Py89j ZGBpEGFgYFBh
gAAHI GZkAj NZFYCEApDnyArhi wKJr2aunmpI cvrK8cnZI HVMDckMLGbpCYb/
...
CCmOj UAEA AHKCAA mA BEAAAAAAAEAI AC2gQAAAABFeHBvcnRTdW1tYXJ5XzIwMTEwMjA3XzFRQTI tMTYOMzhFLnR4dFNEBACKAAAAVVQF
AAFPmN90UESBAhcLFAAAAAgACE6HP1i OKOQbCwAAkVMAAB0AEQAAAAAAQAALaBBQI AAENvbnRhY3RzXzFRQTI tMTYOMzhFXzAwMDEuY
3N2U0QEAKQAAAABVVAUAB8+Y305QSwUGAAAAAAIAAgDBAAAAZA0AAAAA</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 all objects that use Web Services v2.0 Schema and the following objects that use Web Services v1.0 Schema:

Account, Account Competitor, Account Contact, Account Contact Role, Account Note, Account Team, Activity, Activity Attachment, Asset, Call PromItemDrop, Call SampleDropped, Campaign, Contact, Contact Campaign, Contact Interests, Contact Note, Contact Team, Custom Object 01 - 03, Custom Object 01 - 03 Team, Dealer, Event Household, Household Contact, HouseholdTeam, Lead, Lead Team, Opportunity, Opportunity Competitor, Opportunity Contact Role, Opportunity Note, Opportunity Partner, Opportunity Team, Portfolio, Portfolio Owner, PortfolioTeam, Product, ProductCategory, Program, Program Partner, Service Request, Service Request Note, Service Request Team, Solution, Territory, 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`

When company administrators create an Integration Event action on a workflow rule, they can specify the schema version to be used for integration events. The default schema version is Web Services v2.0 schema, but for objects that support both Web Services v1.0 and Web Services v2.0 Schema, the company administrator can select either schema version.

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.

In the Integration Event Queue Settings page in the UI, the Picklist Format field determines whether language-independent values or language-dependent values (in the user's language) for picklists are written to an integration event queue.

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 181](#).

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.”](#)

### Time Zone Setting for Integration Events

On the Integration Event Queue Settings page in the UI your company administrator can specify settings for the default queue. The Time Zone picklist determines how the values of date and time fields (for example, ModifiedDate, CreatedDate) are recorded in integration events.

You can also use the IntegrationEventQueueUpsert method to specify values for the Time Zone field, and the values you can specify are UTC, UTC Format, and User Time Zone (the default value) as explained in [“IntegrationEventQueueUpsert” on page 307](#).

## Arguments

[Table 49](#) describes the arguments taken by the GetEvents method.

Table 49. 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>
              </Contact>
            </ListOfContact>
          </Sibel Message>
        </Event>
      </ListOfEvent>
    </ns:IntegrationEventWS_GetEvents_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
<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>
</SibelMessage>
</Event>
<Event object="Contact" name="cContact Update" operation="update">
  <SibelMessage>
    <ListOfContact xmlns="urn:/crmondemand/xml/contact">
      <Contact>
        <ContactId>ADSA-96D7EA</ContactId>
        <ModId>2</ModId>
        <CreateDate>08/09/2011 10:45:39</CreateDate>
        <ExternalSystemId></ExternalSystemId>
        <ContactFirstName>C</ContactFirstName>
        <IntegrationId>ADSA-96D7EA</IntegrationId>
        <ContactLastName>C8</ContactLastName>
        <ModifiedById>HRZ780-I5RUE</ModifiedById>
        <ModifiedDate>08/09/2011 10:57:52</ModifiedDate>
        <AlternateAddressId>ADSA-96D7EC</AlternateAddressId>
        <CreatedById>HRZ780-I5RUE</CreatedById>
      </Contact>
    </ListOfContact>
  </SibelMessage>
</Event>
<Event object="Account" name="Accounts Create" operation="insert">
  <SibelMessage>
    <ListOfAccount xmlns="urn:/crmondemand/xml/account">
      <Account>
        <AccountId>ADSA-96DJL4</AccountId>
        <ModId>0</ModId>
      </Account>
    </ListOfAccount>
  </SibelMessage>
</Event>
```

```

        <CreatedDate>08/09/2011 11:04:50</CreatedDate>
        <CurrencyCode>USD</CurrencyCode>
        <ExternalSystemId></ExternalSystemId>
        <IntegrationId>ADSA-96DJL4</IntegrationId>
        <Location></Location>
        <ModifiedById>HRZ780-I5RUE</ModifiedById>
        <ModifiedDate>08/09/2011 11:04:51</ModifiedDate>
        <AccountName>A1</AccountName>
        <NumberOfEmployees>1000</NumberOfEmployees>
        <PrimaryBillToAddressId>ADSA-96DJL6</PrimaryBillToAddressId>
        <PrimaryShipToAddressId>ADSA-96DJL8</PrimaryShipToAddressId>
        <CreatedById>HRZ780-I5RUE</CreatedById>
    </Account>
</ListOfAccount>
</SibelMessage>
</Event>
</ListOfEvent>
<ns:LastEventId>20110809160454038_QNMHRZ780-I5RT8_Account_Account_ADSA-96DJL4_0_ADSA-96DCMZ.xml</ns:LastEventId>
</ns:IntegrationEventWS_GetEvents_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

### Sample SOAP Request 2 - GetEvents with EventCount and QueueName

The following SOAP request retrieves one event from the queue High Priority Queue specified by using the <EventCount> and <QueueName> arguments.

```

<?xml version="1.0" encoding="utf-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>
    <wssse:Security>
      <wssse:UsernameToken>
        <wssse:Username>%%USERNAME%%</wssse:Username>
        <wssse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wssse:Password>
      </wssse:UsernameToken>
    </wssse: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>
                <CreatedDate>08/09/2011 10:17:48</CreatedDate>
                <ExternalSystemId></ExternalSystemId>
                <ContactFirstName>Testing</ContactFirstName>
                <IntegrationId>ADSA-96CRZK</IntegrationId>
                <ContactLastName>U2</ContactLastName>
                <ModifiedById>HRZ780-I5RUE</ModifiedById>
                <ModifiedDate>08/09/2011 10:21:16</ModifiedDate>
                <AlternateAddressId>ADSA-96CRZM</AlternateAddressId>
                <CreatedById>HRZ780-I5RUE</CreatedById>
              </Contact>
            </ListOfContact>
          </Siebel Message>
        </Event>
      </ListOfEvent>
    </ns:IntegrationEventWS_GetEvents_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

      <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 Schema file for each record type that you require to track. You can download XSD files for all objects for the Web Services v1.0 API and for the Web Services v2.0 API. 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, one or more `<import>` elements are added to the WSDL file with attributes for the namespace and schema location of an XSD file. The attribute values depend on whether Web Services v1.0 or Web Services v2.0 Schema have been selected for the integration events for the object. The following is an example of an `<import>` element for an object with an integration event for which the Web Services v1.0 Schema is selected:

```
<xsd:import namespace="urn:/crmondemand/xml/account" schemaLocation="Account.xsd" />
```

The following is an example of an `<import>` element for an object with an integration event for which the Web Services v2.0 Schema is selected:

```
<xsd:import namespace="urn:/crmondemand/xml/Account/Data" schemaLocation="Account.xsd" />
```

For more information about specifying the schema version when creating integration event actions, see *Oracle CRM On Demand Online Help*.

### *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.
- 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 system lists and public and private lists are 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 50 describes the arguments taken by the GetLists method:

Table 50. Arguments Taken by the GetLists Method

Name	Description	Required	Default	I/O
ObjectName	The name of the object.	Yes	Not applicable	Input
ListOfLists	The list of lists returned for the object.	Not applicable	Not applicable	Output

## Return Value of the Call

The following information is returned for each list:

- **Name.** The name of the list.
- **ListType.** Whether the list is a system list, 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: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>

    <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>
        <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.

If the record type is supported for Web Services v1.0, then the mapping between the display name of the fields and the XML tag name for Web Services v1.0 is returned. This is the case even if the record type is supported for Web Services v2.0 as well.

If the record type is only supported for Web Services v2.0, then the mapping between the display name and the XML tag name for Web Services v2.0 is returned.

GetMapping returns only the mapping for fields that are exposed for both the UI and Web services. If a field is not exposed in the UI, then it is not returned by GetMapping for the record type.

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 `hi dden`. 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 `hi dden` is returned include the LanguageCode, LocaleCode, RoleId, and TimeZoneld fields of the User object.

## Arguments

[Table 51](#) describes the arguments taken by the GetMapping method.

Table 51. 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 `hi dden` 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: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>
    <wssse:Security>
      <wssse:UsernameToken>
        <wssse:Username>%%USERNAME%%</wssse:Username>
        <wssse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wssse:Password>
      </wssse:UsernameToken>
    </wssse: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>
  <DisplayName>Description</DisplayName>
  <ElementName>Description</ElementName>
  <DataType>Text (Long)</DataType>
</Field>
<Field>
  <DisplayName>Book Name</DisplayName>
```

```

        <ElementName>BookName</ElementName>
        <DataType>Text (Short)</DataType>
    </Field>
</ListOfFields>
</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 but cannot get the lists of possible values for read-only picklist fields.

If the record type is supported for Web Services v1.0, then the picklists returned are those applicable for Web Services v1.0. This is the case even if the record type is supported for Web Services v2.0 as well. If a picklist is not exposed for Web Services v1.0, then the picklist cannot be returned using GetPicklistValues, even if the picklist is exposed for Web Services v2.0.

If the record type is supported only for Web Services v2.0, then the picklists returned are those applicable for Web Services v2.0.

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 52 describes the arguments taken by the GetPicklistValues method.

Table 52. 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: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>
    <wssse:Security>
      <wssse:UsernameToken>
        <wssse:Username>%%USERNAME%%</wssse:Username>
        <wssse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wssse:Password>
      </wssse:UsernameToken>
    </wssse: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>
<ParentFiel dName></ParentFiel dName>
<ParentDisplayValue></ParentDisplayValue>
<ParentCode></ParentCode>
<Display></Display>
<ListOfPicklistValues>
  <PicklistValue>
    <Code>Low</Code>
    <DisplayValue>Faible</DisplayValue>
    <Display>N</Display>
  </PicklistValue>
  <PicklistValue>
    <Code>Medium</Code>
    <DisplayValue>Moyen</DisplayValue>
    <Display>N</Display>
  </PicklistValue>
  <PicklistValue>
    <Code>High</Code>
    <DisplayValue>Élevé</DisplayValue>
    <Display>N</Display>
  </PicklistValue>
</ListOfPicklistValues>
</ListOfParentPicklistValues>
</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>
```

## LeadConvert

Converts leads to accounts, contacts, deal registrations, or opportunities.

### Usage

You use the LeadConvert method to convert leads. To download the Lead Conversion WSDL and to convert leads using Web service calls, the Convert Leads privilege must be enabled for your user role. You can create account, contact, deal registration, and opportunity records, or copy lead information to existing records by converting the lead record. Depending on your company's settings, the new opportunity record can include information that affects revenue calculations.

If the necessary options are available in the lead conversion layouts, you can also do the following:

- Copy the members of the lead team to the team on the account, contact, and opportunity that are created or linked to the lead during the conversion process. You can also determine the record access level to grant to the lead owner in the team.
- Link the lead's associated custom object records to the account, contact, and opportunity that are created or linked to the lead during the conversion process. Custom Objects 01 through 40 are supported.

The LeadConvert method provides the same functionality that is available in the Lead Convert page in the UI. For conversion, a contact record is required, either an existing contact record or a new record.

Table 53 shows how the fields in the Lead Convert UI page map to fields used in LeadConvert Web service requests.

Table 53. Mapping of Lead Convert Page Fields to LeadConvert Fields

UI Field	LeadConvert Method Field	Comments
<b>Account Fields</b>		
Associated Account	AccountName	For existing accounts.  The following fields can be used to pick an existing account: - Id, ExternalSystemId, AccountName and Location.
Account Name	AccountName	For new accounts.
Copy Lead Team	CopyLeadTeamtoAccount	Possible values: true or false, 1 or 0.
Lead Owner Record Access in Account Team	AccountTeamLeadOwner Access	The record access level to grant to the lead owner in the account team.
Associate Custom Object 01 through Associate Custom Object 40	AssociateCO1ToAccount through AssociateC40ToAccount	Possible values: true or false, 1 or 0.

### Contact Fields

Table 53. Mapping of Lead Convert Page Fields to LeadConvert Fields

UI Field	LeadConvert Method Field	Comments
Associated Contact	ContactFirstName ContactLastName	For existing contacts.  The following fields can be used to pick an existing contact: Id, ExternalSystemId, ContactFirstName and ContactLastName.
Contact First Name	ContactFirstName	Required in a conversion request
Contact Last Name	ContactLastName	Required in a conversion request
Copy Lead Team	CopyLeadTeamtoContact	Possible values: true or false, 1 or 0.
Lead Owner Record Access in Contact Team	Contact TeamLeadOwnerAccess	The record access level to grant to the lead owner in the contact team.
Associate Custom Object 01 through Associate Custom Object 40	AssociateCO1ToContact through AssociateC40ToContact	Possible values: true or false, 1 or 0.
<b>Opportunity Fields</b>		
Associated Opportunity	OpportunityName	For new opportunities
Potential Revenue	PotentialRevenue	Also a Deal Registration field
Estimated close date	EstimatedCloseDate	Also a Deal Registration field
Next Step	NextStep	Also a Deal Registration field
Description	Description	Also a Deal Registration field
Copy Lead Team	CopyLeadTeamtoOpportunity	Possible values: true or false, 1 or 0.
Lead Owner Record Access in Opportunity Team	OpportunityTeamLeadOwnerAccess	The record access level to grant to the lead owner in the opportunity team.
Associate Custom Object 01 through Associate Custom Object 40	AssociateCO1ToOppty through AssociateC40ToOppty	Possible values: true or false, 1 or 0.
<b>Deal Registration Fields</b>		
Deal Registration Name	DealRegistrationName	For new opportunities
Principal Partner Account	PartnerName	Required when converting to a deal registration.  Principal Partner Account can be specified using the PartnerName, PartnerId, or PartnerExternalSystemId fields.
Company Name	Company	None

Usually you convert leads that have previously been qualified to opportunities.

You can include up to 20 leads in the request for conversion. You can convert the same lead to an account or contact multiple times, but you can convert a lead to a deal registration or opportunity only once. If you have converted a lead to a deal registration or opportunity, you cannot later convert it to a different deal registration or opportunity. However, a lead that has been converted to a deal registration can also later be converted to a different account or contact.

## Arguments

Table 54 describes the arguments taken by the LeadConvert method.

Table 54. Arguments Taken by the LeadConvert Method

Name	Description	Required	Default	I/O
ListOfLead	The list of leads and associated account, contact, deal registration, and opportunity records.	Yes	Not applicable	Input/Output

## Return Value of the Call

The status keys of the converted lead and the account, contact, opportunity, and deal registration records are returned.

## Sample SOAP Request - LeadConvert

The following SOAP request creates new account, contact, opportunity, and deal registration records for a lead.

```
<?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 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <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>
<LeadConvert_Input xmlns="urn:crmondemand/ws/Leadconvert/">
  <ListOfLead>
    <Lead>
      <LeadFirstName>Lead1</LeadFirstName>
      <LeadLastName>test</LeadLastName>
      <Account>
        <AccountName>My New Opty Account</AccountName>
        <Location>Will not show</Location>
      </Account>
      <Contact>
        <ContactFirstName>Converted</ContactFirstName>
        <ContactLastName>Contact</ContactLastName></Contact>
      <Opportunity>
        <OpportunityName>Converted</OpportunityName>
        <PotentialRevenue>2500</PotentialRevenue>
        <EstimatedCloseDate>2015-12-10T05:00:00Z</EstimatedCloseDate>
        <NextStep>Reconvert</NextStep>
        <Description>Converted Lead - Test</Description>
        <ExternalSystemId>LEAD-CONV-DATA</ExternalSystemId>
      </Opportunity>
      <DealRegistration>
        <DealRegistrationName>DR Converted</DealRegistrationName>
        <Company>Test</Company>
        <PotentialRevenue>2500</PotentialRevenue>
        <EstimatedCloseDate>2015-12-10T05:00:00Z</EstimatedCloseDate>
        <NextStep>Reconvert</NextStep>
        <Description>Converted Lead - Test</Description>
        <ExternalSystemId>LEAD-CONV-DR</ExternalSystemId>
        <PartnerName>test</PartnerName>
    </Lead>
  </ListOfLead>
</LeadConvert_Input>

```

```
        </Deal Registration>
    </Lead>
</ListOfLead>
</LeadConvert_Input>
</soap: Body>
</soap: Envelope>
```

### Sample SOAP Response

The following response contains the status keys of the newly-created 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:LeadConvert_Output xmlns:ns="urn:crmondemand/ws/Leadconvert/">
            <ListOfLead xmlns="urn:/crmondemand/xml/Leadconvert/Data">
                <Lead>
                    <Id>1QA2-2EYBDY</Id>
                    <ExternalSystemId></ExternalSystemId>
                    <ModId>0</ModId>
                    <CreatedById>1-CZ10</CreatedById>
                    <CreatedDate>2015-12-08T07:52:59Z</CreatedDate>
                    <CreatedBy>%%USERNAME%%, 2015-12-08T07:52:59Z</CreatedBy>
                    <ModifiedDate>2015-12-08T07:52:59Z</ModifiedDate>
                    <ModifiedById>1-CZ10</ModifiedById>
                    <ModifiedBy>%%USERNAME%%, 2015-12-08T07:52:59Z</ModifiedBy>
                    <Account>
                        <Id>1QA2-2EXDFQ</Id>
                        <ExternalSystemId></ExternalSystemId>
                        <ModId>0</ModId>
                        <CreatedById>1-CZ10</CreatedById>
```

```

    <CreateDate>2015-12-08T07: 52: 59Z</CreateDate>
    <CreatedBy>%%USERNAM%%, 2015-12-08T07: 52: 59Z</CreatedBy>
    <ModifiedDate>2015-12-08T07: 52: 59Z</ModifiedDate>
    <ModifiedBy>1-CZ10</ModifiedBy>
    <ModifiedBy>%%USERNAM%%, 2015-12-08T07: 52: 59Z</ModifiedBy>
</Account>
<Contact>
    <Id>10A2-R22RU</Id>
    <ExternalSystemId></ExternalSystemId>
    <ModId>0</ModId>
    <CreatedBy>1-CZ10</CreatedBy>
    <CreateDate>2015-12-08T07: 52: 59Z</CreateDate>
    <CreatedBy>%%USERNAM%%, 2015-12-08T07: 52: 59Z</CreatedBy>
    <ModifiedDate>2015-12-08T07: 52: 59Z</ModifiedDate>
    <ModifiedBy>1-CZ10</ModifiedBy>
    <ModifiedBy>%%USERNAM%%, 2015-12-08T07: 52: 59Z</ModifiedBy>
</Contact>
<Opportunity>
    <Id>10A2-2EY4I0</Id>
    <ExternalSystemId></ExternalSystemId>
    <ModId>0</ModId>
    <CreatedBy>1-CZ10</CreatedBy>
    <CreateDate>2015-12-08T07: 52: 59Z</CreateDate>
    <CreatedBy>%%USERNAM%%, 2015-12-08T07: 52: 59Z</CreatedBy>
    <ModifiedDate>2015-12-08T07: 52: 59Z</ModifiedDate>
    <ModifiedBy>1-CZ10</ModifiedBy>
    <ModifiedBy>%%USERNAM%%, 2015-12-08T07: 52: 59Z</ModifiedBy>
</Opportunity>
<DealRegistration>
    <Id>10A2-2EY41J</Id>

```

```
<External SystemId></External SystemId>
<ModId>0</ModId>
<CreatedById>1-CZ10</CreatedById>
<CreatedDate>2015-12-08T07: 52: 59Z</CreatedDate>
<CreatedBy>%%USERNAMER%%, 2015-12-08T07: 52: 59Z</CreatedBy>
<ModifiedDate>2015-12-08T07: 52: 59Z</ModifiedDate>
<ModifiedById>1-CZ10</ModifiedById>
<ModifiedBy>%%USERNAMER%%, 2015-12-08T07: 52: 59Z</ModifiedBy>
</Deal Registration>
</Lead>
</ListOfLead>
</ns: LeadConvert_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 55 describes the arguments taken by the LoginHistoryQueryPage method.

Table 55. Arguments Taken by the LoginHistoryQueryPage Method

Name	Description	Required	Default	I/O
ListOfLoginHistory	The list of object instances queried (input), and after query execution, the list of object instances returned (output).	Yes	Not applicable	Input/Output
PageSize	The maximum number of records displayed on a page following a query.	No	10	Input
StartRowNum	Indicates the row from which the LoginHistoryQueryPage method starts to return records. Use the StartRowNum argument to return a set of records for any given method.  For example, if PageSize=100 and 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.
- **LogoutTimestamp.** The time at which the user last logged out.

### 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>

    <LoginHistoryWS_LoginHistoryQueryPage_Input xmlns="urn:crmondemand/ws/loginhistory/">

      <ListOfLoginHistory>

        <LoginHistory>

          <UserId />

          <FirstName />

          <LastName />

          <UserAlias />

          <LoginName />

          <LoginStatus />

          <LoginTimestamp>&gt;='08/01/2011 00:00:00'</LoginTimestamp>

          <ClientType />

          <IPAddress />

          <AdditionalInformation />

          <LogoutTimestamp />

        </LoginHistory>

      </ListOfLoginHistory>

    </LoginHistoryWS_LoginHistoryQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

```

    </ListOfLogInHistory>
    <PageSize>1</PageSize>
    <StartRowNum>1</StartRowNum>
  </LogInHistoryWS_LogInHistoryQueryPage_Input>
</soap: Body>
</soap: Envelope>

```

## Sample SOAP Response

The following response contains the requested login history:

```

<?xml version="1.0" encoding="utf-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: LogInHistoryWS_LogInHistoryQueryPage_Output xmlns:ns="urn:crmondemand/ws/loginhistory/">
      <ns: LastPage>false</ns: LastPage>
      <ListOfLogInHistory xmlns="urn:/crmondemand/xml/loginhistory">
        <LogInHistory>
          <UserId>HRZ780-15RUE</UserId>
          <FirstName>Joanne</FirstName>
          <LastName>Brown</LastName>
          <UserAlias>Joanne Brown</UserAlias>
          <LoginName>%%USERNAME%%</LoginName>
          <LoginStatus>Success</LoginStatus>
          <LoginTimestamp>08/02/2011 13:53:00</LoginTimestamp>
          <ClientType>WS</ClientType>
          <IPAddress>10.156.87.45</IPAddress>
          <AdditionalInformation></AdditionalInformation>
          <LogoutTimestamp>08/02/2011 13:59:00</LogoutTimestamp>
        </LogInHistory>
      </ListOfLogInHistory>
    </ns: LogInHistoryWS_LogInHistoryQueryPage_Output>
  </SOAP-ENV: Body>
</SOAP-ENV: Envelope>

```

## MergeRecords

Merges records for certain record types.

### Objects Supported

MergeRecords is only supported for the Account, Contact, Division, 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 MergeDuplicateBooks argument together with the Enable Merge of Duplicate Record's Books setting in the company profile determine whether the books of duplicate records are added to the primary record:

- If the company profile setting is not selected, OR if the company profile setting is selected AND the MergeDuplicateBooks argument is not set to true, then the duplicate record's books are not added to the primary record.
- If the company profile setting is selected AND the MergeDuplicateBooks argument is set to true, then the duplicate record's books are added to the primary record.

The Merge Web service has the same security restrictions as in the Oracle CRM On Demand UI regarding privilege and record permissions. The merge operation is only allowed for users who have Delete access to the record type or who have the Merge Records Without Delete Access privilege for their role.

[Table 56 on page 205](#) describes the arguments taken by MergeRecords.

### Possible Reasons for Failure of the Merge Operation

If there are more than 9999 records of any one record type linked to either of the records that you want to merge, then the merge operation will not complete successfully. Also, if there are large numbers of records of any type linked to either of the records that you want to merge, then the merge operation might time out, even if the number of linked records of any one record type does not exceed 9999. Other factors can also cause a merge operation to time out, such as the types of records that are involved in the merge operation, the speed of your network, and the load on the Oracle CRM On Demand database and server. You can work around these issues by doing the following:

- Unlink some of the linked records from the parent record as necessary before you merge the parent records.
- Perform merge operations during off-peak hours.

Table 56. Arguments taken by MergeRecords

Field Name	Description	Required	Default	I/O
PrimaryId	The ID of the primary record.	One of these arguments is required	Not applicable	Input
PrimaryExternalSystemId	The externalsystemID of the primary record.		Not applicable	Input
DuplicateId	The ID of the duplicate record.	One of these arguments is required	Not applicable	Input
DuplicateExternalSystemId	The externalsystemID of the duplicate record.		Not applicable	Input
MergeWhenPrimaryBlank	Determines how records are merged when fields in the primary record are blank.  True values are set as Y, Yes, True, or 1.  False values are any other values including blanks.	No	False	Input
RecordType	The record type; this is case sensitive.	Yes	Not applicable	Input
MergeDuplicateBooks	Determines whether duplicate record's books are retained.	No	False	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 293](#) and [“PicklistUpsert” on page 324](#)).

### Arguments

[Table 57](#) describes the arguments taken by the MetadataChangeSummaryQueryPage method.

Table 57. 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
searchspec	Indicates the search specification.	No	Not applicable	Input
sortsequence	An integer value that determines the order of the sort specification.	No		Input

Table 57. Arguments Taken by the MetadataChangeSummaryQueryPage Method

Name	Description	Required	Default	I/O
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>
  <soap:Body>
```



```

<MetadataChangeSummaryQueryPage_Input xmlns="urn:crmondemand/ws/metadatachangesummary/">
  <ListOfMetadataChangeSummary>
    <MetadataChangeSummary>
      <LOVLastUpdated />
      <CascadeLinksLastUpdated />
      <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>
          <CascadeLinksLastUpdated>2011-08-04T15:06:08</CascadeLinksLastUpdated>
          <FieldManagementLastUpdated>2011-08-03T11:36:21</FieldManagementLastUpdated>
          <WorkflowLastUpdated>2016-01-29T11:32:09</WorkflowLastUpdated>
          <AccessProfileLastUpdated>2016-12-01T20:25:14</AccessProfileLastUpdated>
          <BookOfBusinessLastUpdated>2016-12-19T11:43:06</BookOfBusinessLastUpdated>
        </MetadataChangeSummary>
      </ListOfMetadataChangeSummary>
    </ns:MetadataChangeSummaryQueryPage_Output>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## SalesProcessQueryPage

Returns sales process information for a particular user, the default sales process, or all sales processes for a company.

### Usage

You use the SalesProcessQueryPage method to retrieve sales process information including sales stages, order of the sales stages, default probability, and associated opportunity types.

The ListOfSalesProcess argument contains elements for the various fields of a sales process, and for the associated sales stages and opportunity types. There are child elements ListofSalesStage and ListofOpportunityType.

In requests, if the <SalesProcessId> element does not contain a value, information for all of the sales processes for the company are returned.

If the <SalesProcessId> element contains a value, information for sales processes for the user or role with that sales process ID are returned. You can use the User service to query for the SalesProcId value for a user.

If the <Default> element contains the value Y, information for the default sales process is returned.

For most of the child elements of ListOfSalesProcess, you can use the sortorder and sortsequence attributes to specify how fields in the returned sales process information are sorted.

**NOTE:** The Name and Description fields of SalesProcess do not support the sort specification.

### Arguments

Table 58 describes the arguments taken by the SalesProcessQueryPage method.

Table 58. 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 58. 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 PageSize= 100 and 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 <a href="#">“The LOVLanguageMode Argument” on page 109</a> .	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: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>
```

```

<Sal esProcessQueryPage_I nput xml ns="urn: crmondemand/ws/sal esproc/">
  <Li stOfSal esProcess pagesi ze="1" startrownu m="1" recordcountneeded="true">
    <Sal esProcess>
      <Id />
      <Name />
      <Descri pti on />
      <Defaul t />
      <Li stOfOpportuni tyType pagesi ze="1" startrownu m="0" recordcountneeded="fal se">
        <Opportuni tyType>
          <Id />
          <Type />
        </Opportuni tyType>
      </Li stOfOpportuni tyType>
      <Li stOfSal esStage pagesi ze="5" startrownu m="0" recordcountneeded="true">
        <Sal esStage>
          <Name />
        </Sal esStage>
      </Li stOfSal esStage>
    </Sal esProcess>
  </Li stOfSal esProcess>
</Sal esProcessQueryPage_I nput>
</soap: Body>
</soap: Envel ope>

```

### Sample SOAP Response 1

The following is the response for sample SOAP request 1:

```

<?xml versi on="1.0" encodi ng="utf-8"?>
<SOAP-ENV: Envel ope xml ns: SOAP-ENV="http: //schemas. xml soap. org/soap/envel ope/" xml ns: xsi ="http: //
www. w3. org/2001/XMLSchema-i nstance" xml ns: xsd="http: //www. w3. org/2001/XMLSchema">
  <SOAP-ENV: Body>
    <ns: Sal esProcessQueryPage_Output xml ns: ns="urn: crmondemand/ws/sal esproc/">
      <Li stOfSal esProcess xml ns="urn: /crmondemand/xml /sal esprocess/Data" recordcount="2"
lastpage="true">
        <Sal esProcess>
          <Id>HRZ780-IDI T2</Id>
          <Name>Renewal Sal es Process</Name>
          <Defaul t>N</Defaul t>

```

```

<Description>Simplified process for renewal sales.</Description>
<ListOfOpportunityType Lastpage="true">
  <OpportunityType>
    <Id>HRZ780-IDIV0</Id>
    <Type>Renewal</Type>
  </OpportunityType>
</ListOfOpportunityType>
<ListOfSalesStage recordcount="4" Lastpage="true">
  <SalesStage>
    <Name>Re-Engage</Name>
  </SalesStage>
  <SalesStage>
    <Name>Proposal</Name>
  </SalesStage>
  <SalesStage>
    <Name>Closed/Won</Name>
  </SalesStage>
  <SalesStage>
    <Name>Closed/Lost</Name>
  </SalesStage>
</ListOfSalesStage>
</SalesProcess>
</ListOfSalesProcess>
</ns:SalesProcessQueryPage_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

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

The following SOAP request queries the sales process by specifying the search criteria in the searchspec attribute. It returns all sales processes with a name like Default\* and all child sales stages of the parent sales process with a name like Closed\*. It also demonstrates using the sortorder and sortsequence attributes to sort the result set.

```

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

```

```

    <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>HRZ780-15RS0</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>HRZ780-1DIVS</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 59 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 59. 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>
    <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:Envelope>
```

```
</soap: Header>
<soap: Body>
  <PasswordWS_SetPassword_Input xml ns="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 xml ns: SOAP-ENV="http://schemas.xml soap.org/soap/envelope/" xml ns: xsi="http://
www.w3.org/2001/XMLSchema-instance" xml ns: xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV: Body>
    <ns: PasswordWS_SetPassword_Output xml ns: 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 60 describes the arguments taken by the SetSessionTimeZone method.

Table 60. 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:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%USERNAME%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <TimeWS_SetSessionTimeZone_Input xmlns="urn:crmondemand/ws/time/">
      <TimeZone>(GMT-06:00) Central Time (US & 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 189](#).

### 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.  
**NOTE:** If a value for Order is not included in an update request, the value is not automatically updated. The automatic update functionality is only supported in the UI.
- **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 multi-select picklists. You must create new picklist fields through the Oracle CRM On Demand application UI.

You cannot update read-only picklist fields.

## Arguments

Table 61 describes the arguments taken by the UpdatePicklist method.

Table 61. 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: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>
    <wssse:Security>
      <wssse:UsernameToken>
        <wssse:Username>%%USERNAME%%</wssse:Username>
        <wssse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">%%PASSWORD%%</wssse:Password>
      </wssse:UsernameToken>
    </wssse: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: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>
  <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 189](#).

## 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 62](#) describes the arguments taken by the UpdateCascadingPicklists method.

Table 62. 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 62. 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>
<ParentFieldName>AccountType</ParentFieldName>
<FieldName>Priority</FieldName>
<CreateNew>Y</CreateNew>
<ListOfCascadingPicklistsValue>
  <ParentCascPicklistsValue>
    <ParentCode>Competitor</ParentCode>
    <ParentDisplayValue>Competitor</ParentDisplayValue>
    <ListOfChildPicklistValue>
      <ChildPicklistValue>
        <Code>Low</Code>
      </ChildPicklistValue>
    </ListOfChildPicklistValue>
  </ParentCascPicklistsValue>
  <ParentCascPicklistsValue>
    <ParentCode>Customer</ParentCode>
    <ListOfChildPicklistValue>
      <ChildPicklistValue>
        <Code>Low</Code>
        <DisplayValue>Low</DisplayValue>
      </ChildPicklistValue>
      <ChildPicklistValue>
        <Code>High</Code>
        <DisplayValue>High</DisplayValue>
      </ChildPicklistValue>
    </ListOfChildPicklistValue>
  </ParentCascPicklistsValue>
</ListOfCascadingPicklistsValue>
</PicklistWS_UpdateCascadingPicklistInput>
</soap:Body>
</soap:Envelope>

```

### Sample SOAP Request 2 - UpdateCascadingPicklist: Update an Existing Cascading Picklist

The following SOAP request updates an existing cascading picklist for the account record type. The picklist value is updated in the user's default language:

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

```
<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_UpdateCascadingPickListOutput xmlns:ns="urn:crmondemand/ws/picklist">
      <ns:Status>TRUE</ns:Status>
    </ns:PickListWS_UpdateCascadingPickListOutput>
  </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 63 describes the arguments taken by the UserUsageQueryPage method.

Table 63. 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.
- **ClientName.** The name of the client that made the Web service request. This is the <ClientName> parameter from the SOAP header in Web service requests.
- **ClientType.** The user agent value for the Web service request. For client integrations other than Oracle client integrations, this value defaults to WS.

## 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: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>

    <wssse:Security>

      <wssse:UsernameToken>

        <wssse:Username>%%USERNAME%%</wssse:Username>

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

      </wssse:UsernameToken>

    </wssse: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 />

          <ClientName />

          <ClientType />

        </UserUsage>

      </ListOfUserUsage>

    </UserUsageWS_UserUsageQueryPage_Input>

  </soap:Body>

</soap:Envelope>
```

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

### Sample SOAP Response

The following illustrates the SOAP response. For brevity, a number of <UserUsage> elements are omitted as indicated by the ellipses (...).

```
<?xml version="1.0" encoding="utf-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>
          <ClientName>Acme</ClientName>
          <ClientType>WS</ClientType>
        </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>
    <ClientName>Acme</ClientName>
    <ClientType>WS</ClientType>
  </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 64 on page 234](#). There are the following methods for services, 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. You can update or insert multiple items within an Upsert call.
- **<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* and are also mentioned in specific topics in this chapter. 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 238](#).

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 64. Administrative Services

Service	Method	Usage
Access Profile	<a href="#">AccessProfileRead</a>	Administering access profile configuration data.
	<a href="#">AccessProfileReadAll</a>	
	<a href="#">AccessProfileUpsert</a>	
Action Bar Custom Web Applet	<a href="#">ActionBarCustomWebAppletRead</a>	Administering configuration data for action bar Web applets.  <b>NOTE:</b> Reading or upserting of the Report and Dashboard types of custom Web applet is not supported.
	<a href="#">ActionBarCustomWebAppletReadAll</a>	
	<a href="#">ActionBarCustomWebAppletUpsert</a>	
Action Bar and Global Header Layouts	<a href="#">ActionBarLayoutReadAll</a>	Administering action bar and global header layout configuration data.
	<a href="#">ActionBarLayoutUpsert</a>	
Assessment Scripts	<a href="#">SalesAssessmentTemplateRead</a>	Administering assessment script configuration data.
	<a href="#">SalesAssessmentTemplateReadAll</a>	
	<a href="#">SalesAssessmentTemplateUpsert</a>	
Assignment Rule	<a href="#">AssignmentRuleRead</a>	Administering assignment rule configuration data.
	<a href="#">AssignmentRuleReadAll</a>	
	<a href="#">AssignmentRuleUpsert</a>	
Assignment Rule Group	<a href="#">AssignmentRuleGroupRead</a>	Administering assignment rule group configuration data.
	<a href="#">AssignmentRuleGroupReadAll</a>	
	<a href="#">AssignmentRuleGroupUpsert</a>	
Cascading Picklists	<a href="#">CascadingPicklistRead</a>	Administering cascading picklist configuration data.
	<a href="#">CascadingPicklistReadAll</a>	
	<a href="#">CascadingPicklistUpsert</a>	
Company Profile	<a href="#">CurrentOrganizationRead</a>	Administering the company profile data.

Service	Method	Usage
Concatenated Field	<a href="#">ConcatenatedFieldRead</a>	Administering concatenated field configuration data.
	<a href="#">ConcatenatedFieldReadAll</a>	
	<a href="#">ConcatenatedFieldUpsert</a>	
Currency Picklist	<a href="#">CurrencyRead</a>	Administering currency picklist configuration data.
	<a href="#">CurrencyReadAll</a>	
	<a href="#">CurrencyUpsert</a>	
Custom HTML Head Tag	<a href="#">CustomHTMLHeadTagRead</a>	Administering custom HTML Head tag configuration data.
	<a href="#">CustomHTMLHeadTagUpsert</a>	
Customize Record Types	<a href="#">CustomRecordTypeRead</a>	Administering customized record type configuration data.
	<a href="#">CustomRecordTypeReadAll</a>	
	<a href="#">CustomRecordTypeUpsert</a>	
Custom Web Link	<a href="#">CustomWebLinkRead</a>	Administering Web link configuration data.  <b>NOTE:</b> Reading or upserting of the Report and Dashboard types of custom Web link is not supported.
	<a href="#">CustomWebLinkReadAll</a>	
	<a href="#">CustomWebLinkUpsert</a>	
Custom Web Tab	<a href="#">CustomWebTabDelete</a>	Administering custom Web tab configuration data.  <b>NOTE:</b> Reading or upserting of the Report and Dashboard types of custom Web tab is not supported.
	<a href="#">CustomWebTabRead</a>	
	<a href="#">CustomWebTabReadAll</a>	
	<a href="#">CustomWebTabUpsert</a>	
Detail Page Custom Web Applet	<a href="#">DetailPageCustomWebAppletRead</a>	Administering detail page Web applet configuration data.  <b>NOTE:</b> Reading or upserting of the Report and Dashboard types of custom Web applet is not supported.
	<a href="#">DetailPageCustomWebAppletReadAll</a>	
	<a href="#">DetailPageCustomWebAppletUpsert</a>	
Dynamic Layout	<a href="#">DynamicLayoutRead</a>	Administering dynamic page layout configuration data.
	<a href="#">DynamicLayoutReadAll</a>	
	<a href="#">DynamicLayoutUpsert</a>	
Enable Modification Tracking	<a href="#">EnableModificationTrackingRead</a>	Administering modification tracking configuration data.
	<a href="#">EnableModificationTrackingReadAll</a>	
	<a href="#">EnableModificationTrackingUpsert</a>	

Service	Method	Usage
Field Audit Setup	<a href="#">FieldAuditSetupRead</a>	Administering field audit configuration data.
	<a href="#">FieldAuditSetupReadAll</a>	
	<a href="#">FieldAuditSetupUpsert</a>	
Field Management	<a href="#">FieldManagementCreate</a>	Administering field management configuration data.
	<a href="#">FieldManagementRead</a>	
	<a href="#">FieldManagementReadAll</a>	
	<a href="#">FieldManagementUpsert</a>	
Homepage Layout	<a href="#">HomepageLayoutReadAll</a>	Administering homepage layout configuration data.
	<a href="#">HomepageLayoutUpsert</a>	
Home Page Custom Web Applet	<a href="#">HomepageCustomWebAppletRead</a>	Administering homepage Web applet configuration data. <b>NOTE:</b> Reading or upserting of the Report and Dashboard types of custom Web applet is not supported.
	<a href="#">HomepageCustomWebAppletReadAll</a>	
	<a href="#">HomepageCustomWebAppletUpsert</a>	
Home Tab Custom Web Applet	<a href="#">HomeTabCustomWebAppletRead</a>	Administering My Homepage tab Web applet configuration data. <b>NOTE:</b> Reading or upserting of the Report and Dashboard types of custom Web applet is not supported.
	<a href="#">HomeTabCustomWebAppletReadAll</a>	
	<a href="#">HomeTabCustomWebAppletUpsert</a>	
Industry Picklist	<a href="#">IndustryRead</a>	Administering industry configuration data.
	<a href="#">IndustryReadAll</a>	
Integration Event Queue Management	<a href="#">IntegrationEventQueueRead</a>	Administering integration event queue configuration data.
	<a href="#">IntegrationEventQueueReadAll</a>	
	<a href="#">IntegrationEventQueueUpsert</a>	
Lists	<a href="#">ListRead</a>	Administering list configuration data.
	<a href="#">ListReadAll</a>	
List Access and Order	<a href="#">ListAccessAndOrderRead</a>	Administering list access and order configuration data.
	<a href="#">ListAccessAndOrderReadAll</a>	
	<a href="#">ListAccessAndOrderUpsert</a>	
Page Layout Field	<a href="#">PageLayoutFieldRead</a>	Administering field information for page layout configuration data.
	<a href="#">PageLayoutFieldReadAll</a>	
	<a href="#">PageLayoutFieldUpsert</a>	

Service	Method	Usage
Page Layout Related Information	<a href="#">PageLayoutRelatedInformationRead</a>	Administering related information for page layout configuration data.
	<a href="#">PageLayoutRelatedInformationReadAll</a>	
	<a href="#">PageLayoutRelatedInformationUpsert</a>	
Page Layout Section Translation	<a href="#">PageLayoutSectionReadAll</a>	Administering section information for page layout configuration data.
	<a href="#">PageLayoutSectionUpsert</a>	
Picklist	<a href="#">PicklistRead</a>	Administering picklist configuration data.
	<a href="#">PicklistReadAll</a>	
	<a href="#">PicklistUpsert</a>	
Picklist Value Group	<a href="#">PicklistValueGroupRead</a>	Administering picklist value group configuration data.
	<a href="#">PicklistValueGroupReadAll</a>	
	<a href="#">PicklistValueGroupUpsert</a>	
Process Administration	<a href="#">ProcessAdministrationRead</a>	Administering process administration configuration data.
	<a href="#">ProcessAdministrationReadAll</a>	
	<a href="#">ProcessAdministrationUpsert</a>	
Related Information Layout	<a href="#">RelatedInformationLayoutRead</a>	Administering related information for page layout configuration data.
	<a href="#">RelatedInformationLayoutReadAll</a>	
	<a href="#">RelatedInformationLayoutUpsert</a>	
Report Folders	<a href="#">ReportFolderRead</a>	Administering report and dashboard folder configuration data.
	<a href="#">ReportFolderReadAll</a>	
	<a href="#">ReportFolderUpsert</a>	
Role Management	<a href="#">RoleRead</a>	Administering role management configuration data.
	<a href="#">RoleReadAll</a>	
	<a href="#">RoleUpsert</a>	
Sales Category	<a href="#">SalesCategoryRead</a>	Administering sales category configuration data.
	<a href="#">SalesCategoryReadAll</a>	
	<a href="#">SalesCategoryUpsert</a>	
Search Layout	<a href="#">SearchLayoutRead</a>	Administering search layout configuration data.
	<a href="#">SearchLayoutReadAll</a>	
	<a href="#">SearchLayoutUpsert</a>	
SSO Token	<a href="#">SSOTokenRead</a>	Obtaining a Single-Sign On (SSO) token.

Service	Method	Usage
Workflow Action	<a href="#">WorkflowActionSetRead</a>	Administering workflow action configuration data.
	<a href="#">WorkflowActionSetReadForRuleSet</a>	
	<a href="#">WorkflowActionSetUpsert</a>	
	<a href="#">WorkflowActionSetInsertOrUpdate</a>	
Workflow Rule	<a href="#">WorkflowRuleSetDelete</a>	Administering workflow rule configuration data
	<a href="#">WorkflowRuleSetRead</a>	
	<a href="#">WorkflowRuleSetReadAll</a>	
	<a href="#">WorkflowRuleSetUpsert</a>	
	<a href="#">WorkflowRuleSetInsertOrUpdate</a>	

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).

## Record Ownership Mode

You can use the Field Management service to determine and configure the record ownership mode for objects. There are three record ownership modes for objects: user mode, book mode, and mixed mode. For information about record ownership modes, including which record types support record ownership modes, see *Oracle CRM On Demand Online Help*.

You can determine the record ownership mode by using the [FieldManagementRead](#) method to examine the Required flag for the Owner and Book fields for an object:

- For user mode, the Required flag is true for the Owner field and false for the Book field.
- For book mode, the Required flag is false for the Owner field and true for the Book field.
- For mixed mode, the Required flag is false for both the Owner field and Book field.

If the Required flag is true for both Owner and Book fields, then the configuration is invalid and no records can be saved until the configuration is fixed.

A SOAP request to determine the record ownership mode must specify the object name, and the system names for the Owner and Book fields. It must also specify true for the IncludeAll argument to ensure that the field information is returned. [Table 65](#) shows the system names for the Owner and Book fields for some record types.

Table 65. System Names of Owner and Book Fields for Objects

Object	Owner Field	Book Field
Account	Primary Owner Id	Primary Position Id
Activity	Primary Owner Id	Position Id
Contact	Primary Owner Id	Primary Position Id
Opportunity	Primary Owner Id	Primary Position Id
Lead	Owner Id	Primary Position Id

The following is a sample SOAP request to retrieve the record ownership mode information for the Contact object:

```
<?xml version="1.0" encoding="utf-16"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:fiel="urn:crmondemand/ws/odesabs/fielmanagement/" xmlns:quer="urn:/crmondemand/xml/fielmanagement/query">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-30489657" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>rsbooks/rsinn</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>
  </soapenv:Header>
  <soapenv:Body>
    <fiel:FielManagementRead_Input>
      <quer:FielDataSet>
        <quer:ObjectName>Contact</quer:ObjectName>
        <quer:IncludeAll>true</quer:IncludeAll>
        <quer:ListOfFields>
          <quer:FielId>
            <quer:Name>Primary Owner Id</quer:Name>
            <quer:Required />
          </quer:FielId>
          <quer:FielId>
            <quer:Name>Primary Position Id</quer:Name>
            <quer:Required />
          </quer:FielId>
        </quer:ListOfFields>
      </quer:FielDataSet>
    </fiel:FielManagementRead_Input>
  </soapenv:Body>
</soapenv:Envelope>
```

The following is the SOAP response, and in this case, Required is false for both the Owner and Book fields, which indicates mixed mode:

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



```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header />
  <env:Body>
    <fieldManagementRead_Output xmlns:field="urn:crmondemand/ws/odesabs/fieldmanagement/">
      <data:ListOfFieldSet xmlns:data="urn:/crmondemand/xml/fieldmanagement/data">
        <data:FieldSet>
          <data:ObjectName>Contact</data:ObjectName>
          <data:ListOfFields>
            <data:Field>
              <data:Name>Primary Owner Id</data:Name>
              <data:DisplayName>Primary Owner Id</data:DisplayName>
              <data:Required>false</data:Required>
            </data:Field>
            <data:Field>
              <data:Name>Primary Position Id</data:Name>
              <data:DisplayName>Book: Id</data:DisplayName>
              <data:Required>false</data:Required>
            </data:Field>
          </data:ListOfFields>
        </data:FieldSet>
      </data:ListOfFieldSet>
    </fieldManagementRead_Output>
  </env:Body>
</env:Envelope>

```

You can use the [FieldManagementUpsert](#) method to set the record ownership mode for an object as required.

## AccessProfileRead

Extracts access profile configuration data.

### Usage

Use the `AccessProfileRead` method to extract access profile configuration data matching the input criteria. The Manage Roles and Access privilege is required.

The access profile is specified with the `<Name>` child element of `apQuery:AccessProfile`.

You can use the <NameTranslation> and <AccessProfileId> elements of apQuery:AccessProfile to retrieve access profile configuration data. The <NameTranslation> element specifies the access profile name in the user’s language, and the <AccessProfileId> element specifies an access profile ID.

See “[AccessProfileUpsert](#)” on page 243 for information about the access profile data specified in apData:AccessProfile.

## Arguments

Table 66 describes the arguments taken by the AccessProfileRead method.

Table 66. 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. The Manage Roles and Access privilege is required.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 27, and specified values up to and including 27 return configuration data applicable to Oracle CRM On Demand Release 27.

If the value specified is greater than 27, then all of the configuration data up to the release specified is returned. For example, if a value of 28 is specified, then all of the configuration data in Release 27 plus the newer data up to Release 28 is returned (which includes the <NameTranslation> and <AccessProfileID> elements).

See “[AccessProfileUpsert](#)” on page 243 for information about the access profile data specified in apData:AccessProfile.

## Arguments

Table 67 describes the arguments taken by the AccessProfileReadAll method.

Table 67. Arguments Taken by the AccessProfileReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	27	Input
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 Manage Roles and Access privilege is required.

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.
- **NameTranslation.** The name of the access profile in the user's language. This element is read-only and is ignored for upsert requests.
- **AccessProfileId.** The access profile ID. This element is read-only and is ignored for upsert requests.
- **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.

**NOTE:** The Period object is not supported for the AccessProfileUpsert method in Oracle CRM On Demand Release 23 and later.

## Arguments

Table 68 describes the arguments taken by the AccessProfileUpsert method.

Table 68. 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 246](#) for information about the data specified in apData:ListOfActionBarCustomWebApplet.

## Arguments

[Table 69](#) describes the arguments taken by the `ActionBarCustomWebAppletRead` method.

Table 69. 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.

The input argument `<Version>` specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 27, and specified values up to and including 27 return configuration data applicable to Oracle CRM On Demand Release 27.

If the value specified is greater than 27, then all of the configuration data up to the release specified is returned. For example, if a value of 28 is specified, then all of the configuration data in Release 27 plus the newer data up to Release 28 is returned.

See ["ActionBarCustomWebAppletUpsert" on page 246](#) for information about the data specified in `apData:ListOfActionBarCustomWebApplet`.

## Arguments

Table 70 describes the arguments taken by the `ActionBarCustomWebAppletReadAll` method.

Table 70. Arguments Taken by the `ActionBarCustomWebAppletReadAll` Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	27	Input
<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.

If the Enable Language Translation Support for Web Applets check box is selected in the company profile, then you can use the `<LanguageCode>` and `<DisplayName>` child elements of the `<Translation>` element to insert or update translations of the Web applet display name in each language for your company.

The `apData:ActionBarCustomWebApplet` argument contains the following elements containing action bar Web applet data:

- **DisplayName.** The name of the Web applet. Required.
- **DisplayNameCurrentUserLang.** The name of the Web applet in the language of the currently logged-in user.
- **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.
- **ListOfTranslation.** 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 applet 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.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets.

## Arguments

[Table 71](#) describes the arguments taken by the `ActionBarCustomWebAppletUpsert` method.

Table 71. Arguments Taken by the `ActionBarCustomWebAppletUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:ActionBarCustomWebApplet</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.

## ActionBarLayoutReadAll

Extracts all of the action bar and global header layout configuration data for a company.

## Usage

Use the `ActionBarLayoutReadAll` method to read all of the action bar and global header layouts.

The input argument `<Version>` specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 26, and specified values up to and including 26 return configuration data applicable to Oracle CRM On Demand Release 26.

If the value specified is greater than 26, then all of the configuration data up to the release specified is returned. For example, if a value of 27 is specified, then all of the configuration data in Release 26 plus the newer data up to Release 27 is returned.

See [“ActionBarLayoutUpsert” on page 248](#) for information about the layout data specified in `abldata:ActionBarLayout`.

## Arguments

[Table 72](#) describes the arguments taken by the `ActionBarLayoutReadAll` method.

Table 72. Arguments Taken by the `ActionBarLayoutReadAll` Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	26	Input
<code>abldata:ListOfActionBarLayout</code>	The extracted data for all layouts.	Not applicable	Not applicable	Output

## Return Value of the Call

The layouts are returned.

## ActionBarLayoutUpsert

Updates an existing action bar and global header layout or inserts a new layout.

### Usage

Use the `ActionBarLayoutUpsert` method to insert or update an existing action bar and global header layout.

The `abldata:ActionBarLayout` argument contains the following elements containing layout data:

- **LayoutName.** The name of the layout.
- **Description.** A description of the layout.
- **ListOfSections.** Contains all of the available sections for the layout. The data for a particular section is contained in the following elements:



- **Name.** The name of the section, which can be a custom Web applet configured for the action bar, or 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 and Global Header Layouts wizard in the UI. The integer value can be one of the following:
  - ❑ **-3.** The section is in the Global Header Available Sections list.
  - ❑ **-2.** The section is in the Not Available Sections list.
  - ❑ **-1.** The section is in the Action Bar Available Sections list.
  - ❑ **Positive integers.** The section is in the Action Bar Displayed Sections list. The integer value indicates the location of the section within the list, with 0 (zero) meaning the section is located at the top of the list.
- **GlobalHeaderSection.** Whether the section is a global header section.
- **ListOfGlobalHeaders.** Contains the details for the global header sections, and the data for a global header section is contained in the following elements:
  - **Order.** The order in which the section is displayed in the global header (required).
  - **Icon.** The name of the icon for the section (required).
  - **Applet.** The name of the section (required).
  - **AppletWidth.** The width of the section (required).
  - **AppletDescription.** A description.

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 73 describes the arguments taken by the ActionBarLayoutUpsert method.

Table 73. Arguments Taken by the ActionBarLayoutUpsert Method

Name	Description	Required	Default	I/O
ablData:ActionBarLayout	The 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.

## AssignmentRuleRead

Extracts assignment rule configuration data.

## Usage

Use the AssignmentRuleRead method to extract the assignment rule data matching the input criteria. The Data Rules and Assignment privilege is required.

An assignment rule is specified with the following child elements of apQuery:AssignmentRule:

■ <RuleGroupName> + <RecordType> + <RuleName>

## Arguments

Table 74 describes the arguments taken by the AssignmentRuleRead method.

Table 74. Arguments Taken by the AssignmentRuleRead Method

Name	Description	Required	Default	I/O
apQuery:AssignmentRule	The assignment rule query criteria.	Yes	Not applicable	Input
apData:ListOfAssignmentRule	The extracted assignment rule data.	Not applicable	Not applicable	Output

## Return Value of the Call

The assignment rule data is returned.

## AssignmentRuleReadAll

Extracts all of the assignment rule configuration data for a company.

### Usage

Use the AssignmentRuleReadAll method to read all of the assignment rules. The Data Rules and Assignment privilege is required.

There is no input argument; the method simply returns all of the assignment rule data for each of the supported record types: Account, Lead, Opportunity, and Service Request.

See [“AssignmentRuleUpsert” on page 251](#) for information about the assignment rule data returned in apData:ListOfAssignmentRule.

### Arguments

[Table 75](#) describes the arguments taken by the AssignmentRuleReadAll method.

Table 75. Arguments Taken by the AssignmentRuleReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfAssignmentRule	The extracted data for all assignment rules.	Not applicable	Not applicable	Output

### Return Value of the Call

The assignment rules are returned.

## AssignmentRuleUpsert

Updates assignment rule configuration data.

### Usage

Use the AssignmentRuleUpsert method to insert or update assignment rule data. The Data Rules and Assignment privilege is required.

The apData:AssignmentRule argument contains the following elements specifying the data for the assignment rule:

- **RecordType.** The record type: Account, Lead, Opportunity, or Service Request (read-only).
- **RuleGroupName.** The name of the rule group (read-only).
- **RuleName.** The name of the rule.
- **Order.** The order in which the rules are evaluated.
- **AssignTo.** The user to whom the record is assigned, if the rule criteria are met.

- **SendEmailNotification.** Whether an email notification is sent.
- **AssignToTerritory.** For account and opportunity rules only, the territory to which the record is assigned, if the rule criteria are met.
- **IncludeTeamAssignment.** For account and opportunity rules only, whether team members are assigned to the account, if the rule criteria are met.
- **IncludeRelatedContacts.** For account rules only, whether contacts linked to the account are assigned to the specified account owner and territory if the rule criteria are met.
- **IncludeRelatedOpportunities.** For account rules only, whether opportunities linked to the account are assigned to the specified account owner and territory if the rule criteria are met.
- **ListOfTeam.** Contains the data for team members assigned to the account or opportunity in the following child elements of <Team>:
  - **User.** The first and last name of the team member (read-only).
  - **Role.** The role of the team member (read-only).
  - **AccountAccess.** The access level for accounts.
  - **ContactAccess.** The access level for contacts.
  - **OpportunityAccess.** The access level for opportunities.
- **ListOfRuleCriteria.** Contains the rule criteria for the rule in the following child elements of <Rule>:
  - **Field.** The name of the field to be evaluated.
  - **Condition.** The condition to satisfy to meet the criterion.
  - **Value.** The value for the condition.
  - **Sequence.** The sequence number for the criterion.

You can use the following combinations of fields to identify rules for upsert operations:

- 1 <RuleGroupName> + <RecordType> + <RuleName>
- 2 <RuleGroupName> + <RecordType> + <Order>

If both of these user key combinations have matches, user key 1 takes priority over user key 2.

To identify teams for upsert operations, the user key is the <User> field in <ListOfTeam>.

To identify rule criteria for upsert operations, the user key is the <Sequence> field in <ListOfRuleCriteria>.

For information about assignment rules, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 76 describes the arguments taken by the AssignmentRuleUpsert method.

Table 76. Arguments Taken by the AssignmentRuleUpsert Method

Name	Description	Required	Default	I/O
apData:ListOfAssignmentRule	The assignment 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.

## AssignmentRuleGroupRead

Extracts assignment rule group configuration data.

## Usage

Use the AssignmentRuleGroupRead method to extract the assignment rule group data matching the input criteria. The Data Rules and Assignment privilege is required.

You can read assignment rule data for a particular record type and rule group. The record type is specified with the <RecordType> child element and the rule group is specified with the <RuleGroupName> child element of apQuery:AssignmentRuleGroup.

## Arguments

Table 77 describes the arguments taken by the AssignmentRuleGroupRead method.

Table 77. Arguments Taken by the AssignmentRuleGroupRead Method

Name	Description	Required	Default	I/O
apQuery:AssignmentRuleGroup	The assignment rule group query criteria.	Yes	Not applicable	Input
apData:ListOfAssignmentRuleGroup	The extracted assignment rule group data.	Not applicable	Not applicable	Output

## Return Value of the Call

The assignment rule group data is returned.

## AssignmentRuleGroupReadAll

Extracts all of the assignment rule group configuration data for a company. The Data Rules and Assignment privilege is required.

### Usage

Use the AssignmentRuleGroupReadAll method to read all of the assignment rule groups.

There is no input argument; the method simply returns all of the assignment rule group data for each of the supported record types: Account, Lead, Opportunity, and Service Request.

See [“AssignmentRuleGroupUpsert” on page 254](#) for information about the assignment rule group data returned in apData:ListOfAssignmentRuleGroup.

### Arguments

Table 78 describes the arguments taken by the AssignmentRuleGroupReadAll method.

Table 78. Arguments Taken by the AssignmentRuleGroupReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfAssignmentRuleGroup	The extracted data for all assignment rule groups.	Not applicable	Not applicable	Output

### Return Value of the Call

The assignment rule groups are returned.

## AssignmentRuleGroupUpsert

Updates assignment rule group configuration data.

### Usage

Use the AssignmentRuleGroupUpsert method to insert or update assignment rule group data. The Data Rules and Assignment privilege is required.

The apData:AssignmentRuleGroup argument contains the following elements specifying the data for the assignment rule group:

- **RecordType.** The record type: Account, Lead, Opportunity, or Service Request.
- **RuleGroupName.** The name of the rule group.
- **Active.** Whether the rule group is active.
- **UnassignedOwner.** A user who inherits the records that are not assigned by one of the rules.

- **UnassignedTerritory.** For account and opportunity rule groups only, a territory that inherits the records that are not assigned by one of the rules.
- **ReturnEmail.** For lead and service request rule groups only, the email address that appears in the Sender field in the email notifications that are sent to the owners of leads and service requests.
- **ListOfRule.** Contains all of the rule data, in the following child elements of <AssignmentRule>:
  - **RuleName.** The name of the rule.
  - **Order.** The order in which the rules are evaluated.
  - **AssignTo.** The user to whom the record is assigned, if the rule criteria are met.
  - **EmailNotification.** Whether an email notification is sent to the user to whom the record is assigned. This field is applicable for the Service Request record type only.
  - **AssignToTerritory.** For account and opportunity rules only, the territory to which the record is assigned, if the rule criteria are met.
  - **IncludeTeamAssignment.** For account and opportunity rules only, whether team members are assigned to the account, if the rule criteria are met.
  - **IncludeRelatedContacts.** For account rules only, whether contacts linked to the account are assigned to the specified account owner and territory if the rule criteria are met.
  - **IncludeRelatedOpportunities.** For account rules only, whether opportunities linked to the account are assigned to the specified account owner and territory if the rule criteria are met.

You use <RecordType> + <RuleGroupName> to identify rule groups for upsert operations.

You can use the following combinations of fields to identify rules for upsert operations:

- 1 <RuleGroupName> + <RecordType> + <RuleName>
- 2 <RuleGroupName> + <RecordType> + <Order>

If both of these user key combinations have matches, user key 1 takes priority over user key 2. Upserts without a value for <Order> return an error.

For information about assignment rules, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 79 describes the arguments taken by the AssignmentRuleGroupUpsert method.

Table 79. Arguments Taken by the AssignmentRuleGroupUpsert Method

Name	Description	Required	Default	I/O
apData:AssignmentRuleGroup	The assignment rule group 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.

## 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 80 describes the arguments taken by the CascadingPicklistRead method.

Table 80. 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 256](#) for information about the cascading picklist data returned in `apData:ListOfCascadingPicklistSet`.

### Arguments

[Table 81](#) describes the arguments taken by the CascadingPicklistReadAll method.

Table 81. Arguments Taken by the CascadingPicklistReadAll Method

Name	Description	Required	Default	I/O
<code>apData:ListOfCascadingPicklistSet</code>	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 82 describes the arguments taken by the CascadingPicklistUpsert method.

Table 82. 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 status is returned.

## ConcatenatedFieldRead

Extracts concatenated field configuration data.

### Usage

Use the ConcatenatedFieldRead method to extract concatenated field configuration data for an object.

The object is specified with the <ObjectName> child element of apQuery:ConcatenatedFieldSet.

See [“ConcatenatedFieldUpsert” on page 260](#) for information about the concatenated field configuration data specified in apData:ConcatenatedFieldSet.

## Arguments

[Table 83](#) describes the arguments taken by the ConcatenatedFieldRead method.

Table 83. Arguments Taken by the ConcatenatedFieldRead Method

Name	Description	Required	Default	I/O
apQuery:ConcatenatedFieldSet	The concatenated field query criteria.	Yes	Not applicable	Input
apData:ListOfConcatenatedFieldSet	The extracted concatenated field configuration data.	Not applicable	Not applicable	Output

## Return Value of the Call

The concatenated field configuration data is returned.

## ConcatenatedFieldReadAll

Extracts all of the concatenated field configuration data for a company.

## Usage

Use the ConcatenatedFieldReadAll method to read all of the concatenated field configuration data.

There is no input argument; the method simply returns all of the concatenated field configuration data.

See [“ConcatenatedFieldUpsert” on page 260](#) for information about the concatenated field configuration data specified in apData:ConcatenatedFieldSet.

## Arguments

[Table 84](#) describes the arguments taken by the ConcatenatedFieldReadAll method.

Table 84. Arguments Taken by the ConcatenatedFieldReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfConcatenatedFieldSet	The extracted configuration data for all concatenated fields.	Not applicable	Not applicable	Output

## Return Value of the Call

The concatenated field configuration data is returned.

## ConcatenatedFieldUpsert

Updates concatenated field configuration data or inserts new concatenated field configuration data for an object.

### Usage

Use the `ConcatenatedFieldUpsert` method to insert concatenated field configuration data or update existing concatenated field configuration data for an object.

The object is specified with the `<ObjectName>` child element of `apData:ConcatenatedFieldSet`.

The `apdata:ConcatenatedFieldSet` argument contains the following child elements of `<ConcatenatedField>` containing concatenated field configuration data:

- **Name.** The name of the concatenated field. When performing an insert operation, this name is used as the display name, if a display name is not provided in the `<DisplayName>` element.
- **IntegrationTag.** The field's integration tag name.
- **DisplayText.** The text that is displayed for the concatenated field.
- **ListOfConcatenatedFieldTranslations.** The concatenated fields in the languages activated for your company:
  - **DisplayName.** The display name of the concatenated field in the language identified by the language code.
  - **DisplayText.** The text that is displayed for the concatenated field, 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 concatenated fields, see *Oracle CRM On Demand Online Help*.

## Arguments

[Table 85](#) describes the arguments taken by the ConcatenatedFieldUpsert method.

Table 85. Arguments Taken by the ConcatenatedFieldUpsert Method

Name	Description	Required	Default	I/O
apData:ConcatenatedFieldSet	The concatenated field configuration 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.

## 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 262](#) for information about the currency data specified in apData:ListOfCurrency.

## Arguments

[Table 86](#) describes the arguments taken by the CurrencyRead method.

Table 86. 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 262](#) for information about the currency data specified in apData:ListOfCurrency.

### Arguments

[Table 87](#) describes the arguments taken by the CurrencyReadAll method.

Table 87. 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 88 describes the arguments taken by the CurrencyUpsert method.

Table 88. 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.

The apdata:CurrentOrganization argument contains the elements for company profile data. In the following sections, the elements are grouped according to section name in the Company Profile page in the UI. You can infer the display names of the company profile settings from the associated element names. For information about the company profile settings, see *Oracle CRM On Demand Online Help*.

#### Company Key Information

The elements related to company key information are as follows:

- CompanyName
- SignInId
- Location
- WebSite
- NumberOfEmployees
- MainPhone
- MainFax
- PrimaryContact
- Status

#### Company Contact Information

The elements related to company contact information are as follows:

- BillingCity
- BillingCountry
- BillingPostalCode
- BillingProvince
- BillingAddress1
- BillingAddress2
- ShippingCity
- ShippingCountry
- ShippingPostalCode
- ShippingProvince



- ShippingAddress1
- ShippingAddress2

### Company Settings

The elements related to company settings are as follows:

- DefaultLanguage
- DefaultLocale
- DefaultCurrency
- DefaultTimeZone
- RecordPreviewMode
- InlineEditEnabled
- MessageCenterEnabled
- RecordTypeAuditableFields
- AuditExpiry
- RelatedInfoFormat
- EnhancedViewforTabsEnabled
- ClassicThemePagingforTabsEnabled
- OrderUsage
- DynamicLayoutPersonalizationEnabled
- WebAppletsTranslationEnabled
- WorkflowMessageTranslationEnabled
- GlobalSearchMethod
- FiscalYearStartMonth
- FiscalYearStartDate
- FiscalCalendarType
- ProductProbabilityAveragingEnabled
- SaveAndAddProductEnabled
- OpportunityRevenueSplitEnabled
- HeadUpDisplay
- ListHeaderFreeze
- SharedAddressesEnabled
- ValidateSharedAddresses
- RecordTypeListsInLookupWindows

- ProvincePicklistEnabled
- ExcludeVehicleOrAssetRecordsfromPortfolio
- AutomaticUpdateofOpportunityTotalsEnabled
- UnicodeCharInEmail
- ExportRequestExpiryDays
- ExportRequestAttachmentExpiryDays

### Company Work Week and Calendar Settings

The elements related to work week and calendar settings are as follows:

- Sunday
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- DisplayEventsInCalendar
- CalendarWeekStartDay
- BusinessHoursStartTime
- BusinessHoursEndTime
- ActivityReminder

### Lead Settings

The elements related to lead settings are as follows:

- ConvertOnlyEditableActivities
- OwnerFullNameRatingandSalesPersonarerequiredtoQualifyaLeadEnabled
- AutoUpdateSalesPersonFlag

### Company Theme Settings

The elements related to theme settings are as follows:

- ThemeName
- TabletTheme

### Company Data Visibility Settings

The elements related to data visibility settings are as follows:

- ManagerVisibilityEnabled
- ParentTeamInheritanceforContactEnabled
- ParentTeamInheritanceforOpportunityEnabled
- DefaultGroupAssignment
- BooksEnabled
- BookSelectorEnabled
- BookMergeEnabled
- DisplayParentBookHierarchy

### Integration Settings

The elements related to integration settings are as follows:

- IntegrationEventEnabled
- WorkflowEnabled
- WebServicesR16CompatibilityMode
- IncludeWebLinkURLsInExport

### Company Security Settings

The elements related to company security are as follows:

- CompanyIdleTimeout
- CompanyIdleTimeoutWarning
- AuthenticationType
- ExternalIdentifierforSingleSignOn
- SignInPageforUseridOrPwdAuthentications
- SignInPageforSSOAuthentications
- ITSURLforSSOAuthentications
- CrossSiteRequestForgeryProtectionEnabled
- IFRAMEembeddingEnabled
- CrossSiteScriptingActionCode
- ConvertURLTextValuetoLinksFlag

The `apdata:CurrentOrganization` argument also contains the following elements related to the feature that restricts access to the system to specific IP address ranges:

- **CompIPRestrictionEnabled**. Whether IP Address Restrictions Enabled is set for your company.

- **CompAllowedIP.** The range of IP addresses that are allowed to access the system.
- **ListOfRoleIPRestrictions.** For each user role, the IP address restriction data is contained in the following child elements of <RoleIPRestrictions>:
  - **RoleName.** The name of the role.
  - **RoleIPRestrictionEnabled.** Whether IP Address Restrictions Enabled is set for the role.
  - **RoleAllowedIP.** The range of IP addresses for the role that are allowed to access the system.

For more information about the IP address restriction feature, see *Oracle CRM On Demand Online Help*.

### Analytics Visibility Setting

The elements related to analytics visibility are as follows:

- ReportingSubjectArea
- HistoricalSubjectAreas
- RoleBasedCanReadAllRecords

### Communications Settings

The elements related to communications settings are as follows:

- AutoWrapUp
- SendEmailonUserProfileUpdate

### Additional Information

The elements related to additional information are as follows:

- LicensedUsers
- ActiveUsers
- InactiveUsers
- TrialEndDate
- ModifiedBy

The fields returned include the ActiveUsers and LicensedUsers fields. When administrators add, remove, inactivate, and activate users, or report on monthly user adoption, it is important to know the active and current license counts. The ActiveUsers and LicensedUsers fields allow you to use automated tools to report licensing information.

### Geocoder Settings

The element related to geocoder is as follows:

- GeocoderEnabled

## Arguments

Table 89 describes the arguments taken by the CurrentOrganizationRead method.

Table 89. Arguments Taken by the CurrentOrganizationRead Method

Name	Description	Required	Default	I/O
apData:CurrentOrganization	The company profile data.	Required	Not applicable	Input/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.

## CustomHTMLHeadTagRead

Extracts custom HTML Head tag configuration data.

## Usage

Use the CustomHTMLHeadTagRead method to extract custom HTML Head tag configuration data matching the input criteria. The Upload Client Side Extensions and Manage Custom HTML Head Tag privilege is required.

The HTML head additions are specified with the <HTMLHeadAdditions> element.

For more information about custom HTML head additions, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 90 describes the arguments taken by the CustomHTMLHeadTagRead method.

Table 90. Arguments Taken by the CustomHTMLHeadTagRead Method

Name	Description	Required	Default	I/O
apQuery:CustomHTMLHeadTag	The custom HTML Head tag query criteria.	Yes	Not applicable	Input
apData:CustomHTMLHeadTag	The extracted custom HTML Head tag data.	Not applicable	Not applicable	Output

## Return Value of the Call

The custom HTML Head tag data is returned.

## CustomHTMLHeadTagUpsert

Updates existing custom HTML head additions or inserts new custom HTML head additions.

### Usage

Use the CustomHTMLHeadTagUpsert method to insert or update HTML head additions data. The Upload Client Side Extensions and Manage Custom HTML Head Tag privilege is required.

The apdata:CustomHTMLHeadTag argument contains the following elements containing HTML head additions data: HTMLHeadAdditions.

### Arguments

Table 91 describes the arguments taken by the CustomHTMLHeadTagUpsert method.

Table 91. Arguments Taken by the CustomHTMLHeadTagUpsert Method

Name	Description	Required	Default	I/O
apData:CustomHTMLHeadTag	The HTML head additions 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.

## CustomRecordTypeRead

Extracts customized record type configuration data.

### Usage

Use the CustomRecordTypeRead method to extract customized record type configuration data matching the input criteria.

The 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 272](#) for information about the customized record type data specified in apData:ListOfCustomRecordType.

## Arguments

[Table 92](#) describes the arguments taken by the CustomRecordTypeRead method.

Table 92. Arguments Taken by the CustomRecordTypeRead Method

Name	Description	Required	Default	I/O
apQuery:CustomRecordType	The customized record type query criteria.	Yes	Not applicable	Input
apData:ListOfCustomRecordType	The extracted customized record type data.	Not applicable	Not applicable	Output

## Return Value of the Call

The customized record type data is returned.

## CustomRecordTypeReadAll

Extracts all of the customized record type configuration data for a company.

### Usage

Use the CustomRecordTypeReadAll method to read all of the customized 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.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 21, and specified values up to and including 21 return configuration data applicable to Oracle CRM On Demand Release 21.

If the value specified is greater than 21, then all of the configuration data up to the release specified is returned. For example, if a value of 22 is specified, then all of the configuration data in Release 21 plus the new data in Release 22 is returned.

See [“CustomRecordTypeUpsert” on page 272](#) for information about the customized record type data specified in apData:ListOfCustomRecordType.

## Arguments

Table 93 describes the arguments taken by the CustomRecordTypeReadAll method.

Table 93. 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
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	21	Input
apData:ListOfCustomRecordType	The extracted data for all customized record types.	Not applicable	Not applicable	Output

## Return Value of the Call

The customized record types are returned.

## CustomRecordTypeUpsert

Updates an existing customized record type or inserts a new customized record type.

### Usage

Use the CustomRecordTypeUpsert method to insert or update customized record type data.

The apdata:CustomRecordType argument contains the following elements containing customized record type data:

- **Name.** The object name of the 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.

- **SupportDisplayImage.** Whether an image is displayed on the record type's Detail page. This corresponds to the Display Image check box in the UI.
- **IconName.** The file name of the classic theme icon for the record type.



- **ModernIconName.** The file name of the modern theme icon for the record type.
- **AccessObjName.** The name of the object as used in access profiles (this is read-only).
- **CustomizePreviousOwnerChannel.** The circumstances in which the previous owner of a record is retained in the team when the owner of a record that is shared by a team is changed. This element is applicable only for record types for which sharing of records is supported. The value can be one of the following:
  - **Never Retain Owner.** This is the default value. The previous owner of the record will not be retained in the team.
  - **Always Retain Owner.** The previous owner of the record will be retained in the team, except when the owner is changed using the assignment manager.
  - **All Interactive Operations.** The previous owner of the record will be retained in the team, except when the owner is changed using the Import Assistant, the assignment manager, the mass update feature, or Web services.
- **CustomizePreviousOwnerAccessProfileId.** The ID of the access profile for the previous owner of a shared record.
- **CustomizePreviousOwnerAccessProfile.** The name of the access profile for the previous owner of a shared record.
- **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 customized record type in the language identified by the language code.
  - **PluralName.** The plural name of the customized record type in the language identified by the language code.
  - **ShortName.** The short name of the customized 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 record types in the Oracle CRM On Demand user interface.

**NOTE:** Custom icons may be available for use as modern theme icons, and in this case the filenames used are system-generated. These filenames, for the Icon Type of Mini Icon, are displayed in the Icon Set Detail page in the Oracle CRM On Demand UI. For more information about custom icon sets, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 94 describes the arguments taken by the CustomRecordTypeUpsert method.

Table 94. Arguments Taken by the CustomRecordTypeUpsert Method

Name	Description	Required	Default	I/O
apData:CustomRecordType	The customized 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:WebLinkSet.

See [“CustomWebLinkUpsert” on page 275](#) for information about the custom Web link data specified in apData:WebLinkSet.

**NOTE:** Internally, concatenated fields are custom Web links, however you cannot use the CustomWebLinkRead method to read concatenated field data.

## Arguments

Table 95 describes the arguments taken by the CustomWebLinkRead method.

Table 95. 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 275 for information about the custom Web link data specified in apData:WebLinkSet.

**NOTE:** Internally, concatenated fields are custom Web links, however you cannot use the CustomWebLinkReadAll method to read concatenated field data.

### Arguments

Table 96 describes the arguments taken by the CustomWebLinkReadAll method.

Table 96. 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*.

**NOTE:** Internally, concatenated fields are custom Web links, however you cannot use the CustomWebLinkUpsert method to update or insert concatenated field data.

## Arguments

Table 97 describes the arguments taken by the CustomWebLinkUpsert method.

Table 97. 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 279](#) for information about the custom Web tab data specified in apData:CustomWebTab.

## Arguments

Table 98 describes the arguments taken by the CustomWebTabDelete method.

Table 98. 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 Manage Roles and Access privilege is required.

The custom Web tab is specified with the <DisplayName> child element of apQuery:CustomWebTab.

See [“CustomWebTabUpsert” on page 279](#) for information about the custom Web tab data specified in apData:CustomWebTab.

### Arguments

[Table 99](#) describes the arguments taken by the CustomWebTabRead method.

Table 99. 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 279](#) for information about the custom Web tab data specified in apData:CustomWebTab.

## Arguments

Table 100 describes the arguments taken by the CustomWebTabReadAll method.

Table 100. 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 101](#) describes the arguments taken by the CustomWebTabUpsert method.

Table 101. 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 282](#) for information about the data specified in apData:ListOfDetailPageCustomWebApplet.



## Arguments

[Table 102](#) describes the arguments taken by the DetailPageCustomWebAppletRead method.

Table 102. Arguments Taken by the DetailPageCustomWebAppletRead Method

Name	Description	Required	Default	I/O
apQuery:DetailPageCustomWebApplet	The Web applet query criteria.	Yes	Not applicable	Input
apData:ListOfDetailPageCustomWebApplet	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.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 27, and specified values up to and including 27 return configuration data applicable to Oracle CRM On Demand Release 27.

If the value specified is greater than 27, then all of the configuration data up to the release specified is returned. For example, if a value of 28 is specified, then all of the configuration data in Release 27 plus the newer data up to Release 28 is returned.

See [“DetailPageCustomWebAppletUpsert” on page 282](#) for information about the data specified in apData:ListOfDetailPageCustomWebApplet.

## Arguments

Table 103 describes the arguments taken by the DetailPageCustomWebAppletReadAll method.

Table 103. Arguments Taken by the DetailPageCustomWebAppletReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	27	Input
apData:ListOfDetailPageCustomWebApplet	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.

If the Enable Language Translation Support for Web Applets check box is selected in the company profile, then you can use the <LanguageCode> and <DisplayName> child elements of the <Translation> element to insert or update translations of the Web applet display name in each language for your company.

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.
- **DisplayNameCurrentUserLang.** The name of the Web applet in the language of the currently logged-in user.
- **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.
- **ModernIconName.** The file name of the modern theme icon for the Web applet.
- **ListOfTranslation.** 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 applet 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.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets.

## Arguments

Table 104 describes the arguments taken by the `DetailPageCustomWebAppletUpsert` method.

Table 104. 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.

## DynamicLayoutRead

Extracts all of the configuration data for a dynamic page layout for an object.

## Usage

Use the `DynamicLayoutRead` method to read the dynamic page layout data for an object.

The object is specified with the `<ObjectName>` child element of `<apQuery:DynamicLayoutSet>`.

See [“DynamicLayoutUpsert” on page 285](#) for information about the dynamic page layout data specified in `apData:ListOfDynamicLayoutSet`.

## Arguments

[Table 105](#) describes the arguments taken by the `DynamicLayoutRead` method.

Table 105. Arguments Taken by the `DynamicLayoutRead` Method

Name	Description	Required	Default	I/O
<code>apQuery:DynamicLayoutSet</code>	The dynamic page layout query criteria.	Yes	Not applicable	Input
<code>apData:ListOfDynamicLayoutSet</code>	The extracted dynamic page layout data.	Not applicable	Not applicable	Output

## Return Value of the Call

The dynamic page layout is returned.

## DynamicLayoutReadAll

Extracts all of the dynamic page layout configuration data for a company.

## Usage

Use the `DynamicLayoutReadAll` method to read all of the dynamic page layout data.

There is no input argument; the method simply returns all of the dynamic page layout data for all record types.

See [“DynamicLayoutUpsert” on page 285](#) for information about the dynamic page layout data specified in `apData:ListOfDynamicLayoutSet`.

## Arguments

Table 106 describes the arguments taken by the DynamicLayoutReadAll method.

Table 106. Arguments Taken by the DynamicLayoutReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfDynamicLayoutSet	The extracted dynamic page layout data.	Not applicable	Not applicable	Output

## Return Value of the Call

The dynamic page layouts are returned.

## DynamicLayoutUpsert

Updates existing dynamic page layout data or inserts new dynamic page layout data.

### Usage

Use the DynamicLayoutUpsert method to insert or update dynamic page layout data.

The apdata:DynamicLayoutSet argument contains the following elements containing dynamic page layout data:

- **ObjectName.** The name of the object.
- **DrivingPicklist.** The driving picklist.
- **ListOfDynamicLayouts.** Contains all of the dynamic page layout data. The data for a dynamic page layout is contained in the following child elements of <DynamicLayout>:
  - **Name.** The name of the dynamic page layout.
  - **DefaultLayout.** The default page layout with which driving picklist values are associated.
  - **Description.** A description of the dynamic page layout.
  - **ListOfPicklistValueMapping.** The mappings of driving picklist values to page layouts.
    - **Value.** A driving picklist value.
    - **AssignedLayout.** The page layout that is displayed when the driving picklist value is selected.

See *Oracle Migration Tool On Demand Guide* for reference tables about the mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

## Arguments

Table 107 describes the arguments taken by the DynamicLayoutUpsert method.

Table 107. Arguments Taken by the DynamicLayoutUpsert Method

Name	Description	Required	Default	I/O
apData:DynamicLayoutSet	The dynamic page 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.

## EnableModificationTrackingRead

Extracts modification tracking configuration data.

## Usage

Use the EnableModificationTrackingRead method to extract modification tracking configuration data for an object. The Manage Modification Tracking privilege is required.

The object is specified with the <Object> child element of apQuery:EnableModificationTracking. The <Enabled> child element of the returned apData:EnableModificationTracking indicates whether the object is enabled for modification tracking.

## Arguments

Table 108 describes the arguments taken by the EnableModificationTrackingRead method.

Table 108. Arguments Taken by the EnableModificationTrackingRead Method

Name	Description	Required	Default	I/O
apQuery:EnableModificationTracking	The modification tracking query criteria.	Yes	Not applicable	Input
apData:ListOfEnableModificationTracking	The extracted modification tracking data.	Not applicable	Not applicable	Output

## Return Value of the Call

The modification tracking data is returned.

## EnableModificationTrackingReadAll

Extracts all of the modification tracking configuration data for a company.

### Usage

Use the EnableModificationTrackingReadAll method to read all of the modification tracking data. The Manage Modification Tracking privilege is required.

The <Enabled> child element of the returned apData:ListofEnableModificationTracking indicates whether each object is enabled for modification tracking.

### Arguments

Table 109 describes the arguments taken by the EnableModificationTrackingReadAll method.

Table 109. Arguments Taken by the EnableModificationTrackingReadAll Method

Name	Description	Required	Default	I/O
apData:ListofEnableModificationTracking	The extracted modification tracking data.	Not applicable	Not applicable	Output

### Return Value of the Call

The modification tracking data is returned.

## EnableModificationTrackingUpsert

Updates existing modification tracking data or inserts modification tracking configuration data for an object.

### Usage

Use the EnableModificationTrackingUpsert method to enable or disable modification tracking for objects. The Manage Modification Tracking and Manage Roles and Access privileges are required.

The object is specified with the <Object> child element of apData:EnableModificationTracking, and the <Enabled> element specifies whether the object is enabled (true) or disabled (false) for modification tracking.

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.

For more information about modification tracking setup, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 110 describes the arguments taken by the EnableModificationTrackingUpsert method.

Table 110. Arguments Taken by the EnableModificationTrackingUpsert Method

Name	Description	Required	Default	I/O
apData:ListOfEnableModificationTracking	The modification tracking 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.

## FieldAuditSetupRead

Extracts field audit configuration data.

## Usage

Use the FieldAuditSetupRead method to extract field audit configuration data for an object. The Administer Field Audit privilege is required.

The object is specified with the <ObjectName> child element of apQuery:FieldSet.

See “[FieldAuditSetupUpsert](#)” on page 289 for information about the field audit data specified in apData:FieldSet.

## Arguments

Table 111 describes the arguments taken by the FieldAuditSetupRead method.

Table 111. Arguments Taken by the FieldAuditSetupRead Method

Name	Description	Required	Default	I/O
apQuery:FieldSet	The field audit query criteria.	Yes	Not applicable	Input
apData:ListOfFieldSet	The extracted field audit data.	Not applicable	Not applicable	Output

## Return Value of the Call

The field audit data is returned.



## FieldAuditSetupReadAll

Extracts all of the field audit configuration data for a company.

### Usage

Use the FieldAuditSetupReadAll method to read all of the field audit data. The Administer Field Audit privilege is required.

See “FieldAuditSetupUpsert” on page 289 for information about the field audit data specified in apData:FieldSet.

### Arguments

Table 112 describes the arguments taken by the FieldAuditSetupReadAll method.

Table 112. Arguments Taken by the FieldAuditSetupReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfFieldSet	The extracted field audit data.	Not applicable	Not applicable	Output

### Return Value of the Call

The field audit data is returned.

## FieldAuditSetupUpsert

Updates existing field audit data or inserts field audit configuration data for an object.

### Usage

Use the FieldAuditSetupUpsert method to insert field audit data or update existing field audit data for an object. The Administer Field Audit privilege is required.

The object is specified with the <ObjectName> child element of apData:FieldSet.

The apdata:FieldSet argument contains the following elements containing field audit data:

- **ObjectName.** The system name of the object.
- **AuditRecordCreation.** Whether the creation of records is tracked.
- **AuditRecordDeletionOrRestore.** Whether the deletion or restoration of records is tracked.
- **ListOfFields.** The list of available fields for the object.
  - **Name.** The system name of the field.
  - **IsAudited.** Whether the field is audited.

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.

For more information about field audit setup, see *Oracle CRM On Demand Online Help*.

### Arguments

[Table 113](#) describes the arguments taken by the `FieldAuditSetupUpsert` method.

Table 113. Arguments Taken by the `FieldAuditSetupUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:FieldSet</code>	The field audit 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 custom fields.

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 293](#).

By specifying true for the `<Optimized>` element in the `apData:CustomField` argument, you can create an optimized custom field.

Several record types do not support regular custom fields of particular types, but do support optimized custom fields. These record types include:

- Sales Assessment Template
- Product Indication
- Opportunity Team
- Account Team

- Lead Team
- Business plan Team
- Service Request Team

For example, Sales Assessment Template does not support regular custom Text (Long) fields but it does support optimized custom Text (Long) fields. For Web service requests to create such fields, you must explicitly specify true for <Optimized>, otherwise an error is generated. In the UI, the Optimized check box will be selected and disabled for such record types. As you cannot deselect the Optimized check box, you are forced to create the field as optimized.

## Arguments

Table 114 describes the arguments taken by the FieldManagementCreate method.

Table 114. Arguments Taken by the FieldManagementCreate Method

Name	Description	Required	Default	I/O
apData:CustomField	The field management query criteria.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

## Return Value of the Call

The status of the call is returned.

## FieldManagementRead

Extracts field management configuration data.

### Usage

Use the FieldManagementRead method to extract field management configuration data for an object.

The object is specified with the <ObjectName> child element of apQuery:FieldSet.

If the <IncludeAll> child element of apQuery:FieldSet specifies true, the method returns configuration data including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

Regardless of whether only customized data is returned, or preconfigured and customized data, translated data for all languages enabled by the company are included in the output.

See “[FieldManagementUpsert](#)” on page 293 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.

**NOTE:** Configuration data is returned for the Call Activity History record type even though it is not exposed in the Oracle CRM On Demand UI.

## Arguments

Table 115 describes the arguments taken by the FieldManagementRead method.

Table 115. 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.

When the input argument <IncludeAll> and DefaultAddressField element of apData:FieldSet specify true, the method returns all preconfigured and customized field data, including default address fields.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 23, and specified values up to and including 23 return configuration data applicable to Oracle CRM On Demand Release 23.

If the value specified is greater than 23, then all of the configuration data up to the release specified is returned. For example, if a value of 25 is specified, then all of the configuration data in Release 23 plus the newer data up to Release 25 is returned.

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 293 for information about the field management data specified in apData:FieldSet.

**NOTE:** Configuration data is returned for the Call Activity History record type although the record type is not supported for insert or update operations. The Call Activity History record type does not have a tab in the Oracle CRM On Demand UI, but it is available for field management in the UI.

## Arguments

Table 116 describes the arguments taken by the FieldManagementReadAll method.

Table 116. Arguments Taken by the FieldManagementReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	23	Input
apData:ListOfFieldSet	The extracted field management data.	Not applicable	Not applicable	Output

## Return Value of the Call

The field management data is returned.

## FieldManagementUpsert

Updates existing field management data or inserts field management data for an object.

### Usage

Use the FieldManagementUpsert method to insert field management data or update existing field management data for an object.

The object is specified with the <ObjectName> child element of apData:FieldSet.

The apdata:FieldSet argument contains the following elements containing field management data:

- **Name.** The system name of the field.
- **DisplayName.** The name of the field used in the UI. When performing an insert operation, this name is used as the display name, if a display name is not provided in the <FieldManagementTranslation> element.

- **FieldType.** The type of field.
- **IntegrationTag.** The field's custom integration tag name. This tag can be modified by the user.
- **GenericIntegrationTag.** The field's default integration tag name assigned by Oracle CRM On Demand. This tag cannot be modified by the user. The default integration tag name has the format *fieldtypeDisplay\_Name*. For example, *stMy\_Custom\_Field* would be assigned to a custom field of type Text (Short) with the display name of My Custom Field.

**NOTE:** For 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).
- **RequiredCondition.** An expression used to determine whether the field is required. If the `<Required>` element is set to true, any value in the `<RequiredCondition>` element produces an error message. For more information about conditionally-required fields, see *Oracle CRM On Demand Online Help*.
- **ReadOnly.** Whether the field is a read-only field (true), or is not a read-only field (false).
- **Optimized.** Whether the field is an optimized custom field. You cannot update this value.
- **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.
- **DenormalizedControl.** This maps to the Related Information check box in field management.

For Custom Object record types and certain other record types, you can create custom related information fields. For these record types, the following fields are available on the Field Edit page in the UI: Related Information, Related Record Type, and Related Field. These fields are used only for creating custom related information fields. Most of the other fields on the Field Edit page become read-only fields after you select the Related Information check box. For more information about custom related information fields, see *Oracle CRM On Demand Online Help*.
- **DenormalizedAssociateControlName.** This maps to the Related Record Type picklist in field management.
- **DenormalizedControlName.** This maps to the Related Field picklist in field management.
- **ValidationErrorMsg.** The custom error message displayed if the field fails field validation.
- **TooltipFormat.** The format of the tooltip text, which can be HTML or Plain Text.
- **TooltipText.** The tooltip text that is displayed for the field in the UI.
- **Description.** Additional information about the field.
- **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.
- **TooltipText.** The tooltip text for the field in the language identified by the language code.
- **Description.** The description of the field 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 117](#) describes the arguments taken by the FieldManagementUpsert method.

Table 117. 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 296](#) for information about the homepage layout data specified in apData:ListOfHomepageLayout.

## Arguments

Table 118 describes the arguments taken by the `HomepageLayoutReadAll` method.

Table 118. Arguments Taken by the `HomepageLayoutReadAll` Method

Name	Description	Required	Default	I/O
<code>apData:ListOfHomepageLayout</code>	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.** An internal applet name that is associated with `<SectionLayoutName>` (read only).
  - **SectionLayoutName.** The name of a homepage section in the layout. This name is not the same as the display name of the section as it appears in the Homepage Layout wizard.
- **ListOfSelectedSection.** Contains all of the selected sections for the homepage layout. The data for a particular section is contained in the following elements:
  - **SectionName.** An internal applet name that is associated with `<SectionLayoutName>` (read only).
  - **SectionLayoutName.** The name of a homepage section in the layout. This name is not the same as the display name of the section as it appears in the Homepage Layout wizard.
  - **Sequence.** The sequence of the section within the lists of sections:
    - 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 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 119 describes the arguments taken by the `HomepageLayoutUpsert` method.

Table 119. Arguments Taken by the `HomepageLayoutUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:HomepageLayout</code>	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 299](#) for information about the data specified in `apData:ListOfHomePageCustomWebApplet`.

## Arguments

Table 120 describes the arguments taken by the `HomepageCustomWebAppletRead` method.

Table 120. Arguments Taken by the `HomepageCustomWebAppletRead` Method

Name	Description	Required	Default	I/O
<code>apQuery:HomePageCustomWebApplet</code>	The Web applet query criteria.	Yes	Not applicable	Input
<code>apData:ListOfHomePageCustomWebApplet</code>	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.

The input argument `<Version>` specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 27, and specified values up to and including 27 return configuration data applicable to Oracle CRM On Demand Release 27.

If the value specified is greater than 27, then all of the configuration data up to the release specified is returned. For example, if a value of 28 is specified, then all of the configuration data in Release 27 plus the newer data up to Release 28 is returned.

See [“HomepageCustomWebAppletUpsert” on page 299](#) for information about the data specified in `apData:ListOfHomePageCustomWebApplet`.

## Arguments

Table 121 describes the arguments taken by the HomepageCustomWebAppletReadAll method.

Table 121. Arguments Taken by the HomepageCustomWebAppletReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	27	Input
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.

If the Enable Language Translation Support for Web Applets check box is selected in the company profile, then you can use the <LanguageCode> and <DisplayName> child elements of the <Translation> element to insert or update translations of the Web applet display name in each language for your company.

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.
- **DisplayNameCurrentUserLang.** The name of the Web applet in the language of the currently logged-in user.
- **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.
- NOTE:** Reports in the V3 catalog are not supported.
- **ListOfTranslation.** 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 applet 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.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets and custom homepage reports.

## Arguments

Table 122 describes the arguments taken by the `HomepageCustomWebAppletUpsert` method.

Table 122. Arguments Taken by the `HomepageCustomWebAppletUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:HomePageCustomWebApplet</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.

## 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 302 for information about the data specified in apData:ListOfHomeTabCustomWebApplet.

### Arguments

Table 123 describes the arguments taken by the HomeTabCustomWebAppletRead method.

Table 123. 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.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 27, and specified values up to and including 27 return configuration data applicable to Oracle CRM On Demand Release 27.

If the value specified is greater than 27, then all of the configuration data up to the release specified is returned. For example, if a value of 28 is specified, then all of the configuration data in Release 27 plus the newer data up to Release 28 is returned.

See “[HomeTabCustomWebAppletUpsert](#)” on page 302 for information about the data specified in apData:ListOfHomeTabCustomWebApplet.

## Arguments

Table 124 describes the arguments taken by the HomeTabCustomWebAppletReadAll method.

Table 124. Arguments Taken by the HomeTabCustomWebAppletReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	27	Input
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.

If the Enable Language Translation Support for Web Applets check box is selected in the company profile, then you can use the <LanguageCode> and <DisplayName> child elements of the <Translation> element to insert or update translations of the Web applet display name in each language for your company.

The apData:HomeTabCustomWebApplet argument contains the following elements containing My Homepage tab Web applet data:

- **DisplayName.** The name of the Web applet. Required.
- **DisplayNameCurrentUserLang.** The name of the Web applet in the language of the currently logged-in user.
- **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.
- NOTE:** Reports in the V3 catalog are not supported.
- **ListOfTranslation.** 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 applet 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.

See *Oracle CRM On Demand Online Help* for more information about creating Web applets and custom reports.

## Arguments

Table 125 describes the arguments taken by the HomeTabCustomWebAppletUpsert method.

Table 125. 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 304](#) for information about the industry data specified in apData: Industry.

### Arguments

[Table 126](#) describes the arguments taken by the IndustryRead method.

Table 126. 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.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 22, and specified values up to and including 22 return configuration data applicable to Oracle CRM On Demand Release 22.



If the value specified is greater than 22, then all of the configuration data up to the release specified is returned. For example, if a value of 23 is specified, then all of the configuration data in Release 22 plus the new data in Release 23 is returned.

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.
- **Id.** The ID of the industry record (read-only).

## Arguments

Table 127 describes the arguments taken by the IndustryReadAll method.

Table 127. Arguments Taken by the IndustryReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	22	Input
apData:ListOfIndustry	The extracted industry data for the company.	Not applicable	Not applicable	Output

## Return Value of the Call

The industry data for the company is returned.

## IntegrationEventQueueRead

Extracts all of the configuration data for an integration event queue.

## Usage

Use the `IntegrationEventQueueRead` method to read the configuration data for an integration event queue.

The integration event queue is specified with the `<QueueName>` child element of `apQuery: IntegrationEventQueue`.

See [“IntegrationEventQueueUpsert” on page 307](#) for information about the data specified in `apData: IntegrationEventQueue`.

## Arguments

[Table 128](#) describes the arguments taken by the `IntegrationEventQueueRead` method.

Table 128. Arguments Taken by the `IntegrationEventQueueRead` Method

Name	Description	Required	Default	I/O
<code>apQuery: IntegrationEventQueue</code>	The integration event queue query criteria.	Yes	Not applicable	Input
<code>apData: IntegrationEventQueue</code>	The extracted integration event queue data.	Not applicable	Not applicable	Output

## Return Value of the Call

The integration event queue configuration data is returned.

## IntegrationEventQueueReadAll

Extracts all of the integration event queue configuration data for a company.

## Usage

Use the `IntegrationEventQueueReadAll` method to read all of the integration event queue configuration data for all queues.

See [“IntegrationEventQueueUpsert” on page 307](#) for information about the integration event queue configuration data specified in `apData: IntegrationEventQueue`.

## Arguments

Table 129 describes the arguments taken by the IntegrationEventQueueReadAll method.

Table 129. Arguments Taken by the IntegrationEventQueueReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfIntegrationEventQueue	The extracted integration event queue data.	Not applicable	Not applicable	Output

## Return Value of the Call

The integration event queue configuration data is returned.

## IntegrationEventQueueUpsert

Updates existing integration event queue data or inserts new queue data.

### Usage

Use the IntegrationEventQueueUpsert method to insert or update integration event queue configuration data. The Manage Company privilege is required.

You can use the IntegrationEventQueueUpsert method to clear transactions from an integration event queue. You can clear all transactions from the queue by specifying the value true for the <ClearAllTransactions> element, or you can clear older transactions from the queue by specifying a date value in the <ClearFromDate> element. The upsert operation then clears all of the transactions that occurred before that date from the queue.

The apdata: IntegrationEventQueue argument contains the following elements containing integration event queue data:

- **QueueName.** The name of the queue (required).
- **Description.** A description of the usage of the queue.
- **Disabled.** Whether the queue is disabled.
- **Created.** The username and time and date for the user who created the queue (read-only).
- **Modified.** The username and time and date for the user who last updated the queue (read-only).
- **FileLimit.** The total number of events that the company can store for all queues. If the size of the existing queues matches this number, then you cannot create a new queue. This element is read-only and is ignored for upsert requests.
- **UnassignedQueueCapacity.** The available size of queue. This element is read-only and is ignored for upsert requests.
- **QueueSize.** The size of the queue. This size cannot exceed the <UnassignedQueueCapacity> value. Required for insert operations.

- **NumberOfTransactions.** The number of transactions in the queue. This element is read-only and is ignored for upsert requests.
  - **ClearAllTransactions.** Whether all transactions in the queue should be deleted.
  - **ClearFromDate.** A date and time for which all transactions older than the specified date should be deleted. This must be specified in the `xsd:dateTime` format.
  - **NotificationEmailAddress.** An email address to which a warning email is to be sent when the queue is full or when the queue reaches a certain size, as specified in `<WarningMailQueueSize>`.
  - **WarningMailQueueSize.** The size that the queue is allowed to reach before a warning email is sent to the address specified in `<NotificationEmailAddress>`.
  - **TimeZone.** A value that specifies how the `<Created>` and `<Modified>` values are recorded in the integration events.
    - **UTC.** If this value is specified, then the date and time in the GMT time zone are recorded, in the following format:  
`MM/DD/YYYY HH:MM:SS`  
For example, `06/12/2017 09:44:15`, which corresponds to June 12th 2017, 9:44:15 GMT.
    - **UTC Format.** If this value is specified, then the date and time in the GMT time zone are recorded, in the `xsd:dateTime` format specified by the W3C standard for Coordinated Universal Time (UCT), as follows:  
`YYYY-MM-DDTHH:MM:SSZ`  
For example, `2017-06-12T09:44:15Z`, which corresponds to June 12th 2017, 9:44:15 GMT.
    - **User Time Zone.** If this value is specified, then the date and time in the time zone of the user who updated the record are recorded, in the following format:  
`MM/DD/YYYY HH:MM:SS`  
This value is the default.
- NOTE:** You set the `<TimeZone>` field for the default queue only. All other queues inherit the setting that is selected for the company's default queue. The `<TimeZone>` field is returned for all queues, but it is read-only for queues other than the default queue.
- **PicklistFormat.** The format for recording picklist field values in the integration events, as follows:
    - If you want the values to be recorded as language-independent code (LIC) values, then specify Language Independent Values.
    - If you want the values to be recorded in the language of the user whose action causes the integration event to be created, then specify User Language.
- This element is required for insert operations, and read-only for update operations.

## Arguments

Table 130 describes the arguments taken by the IntegrationEventQueueUpsert method.

Table 130. Arguments Taken by the IntegrationEventQueueUpsert Method

Name	Description	Required	Default	I/O
apData: IntegrationEvent Queue	The integration event queue 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.

## ListRead

Extracts list configuration data.

## Usage

Use the ListRead method to read list configuration data.

You can use the following elements of apQuery:List to query lists:

- <RecordType>
- <ListName>
- <ListType>
- <CreatedByAlias>
- <ModifiedByAlias>

You can read the following types of list:

- **Public lists.** No privileges are required to read public lists.
- **Private lists.** To read private lists defined by all users in the company, the Manage Private Lists privilege is required. No privilege is required to read private lists created by the currently logged in user.
- **Role-specific lists.** To read all role-specific lists defined for the company, the Manage Public Lists privilege is required.

■ **System lists.** To return system lists you can specify System for <ListType>. Otherwise, if the <IncludeAll> child element of apQuery:List specifies true, the ListRead method returns all lists including system lists. If <IncludeAll> specifies false, or is not included in the request, the method returns all configuration data except system list data.

**NOTE:** If the ListRead method is invoked with any search fields other than <RecordType>, or <ListType> with a value other than System, then even if <IncludeAll> is true, the system lists are not returned.

Table 131 shows which lists you can read when you have the Manage Public Lists or Manage Private Lists privilege:

Table 131. Privileges and Access to Lists

Manage Public Lists	Manage Private Lists	Private Lists	Role-specific Lists	Public Lists
Yes	Yes	All	All	All
Yes	No	Private lists created by currently logged in user only	All	All
No	Yes	All	Lists for currently logged in user's role only	All
No	No	Private lists created by currently logged in user only	Lists for currently logged in user's role only	All

**NOTE:** Lists for the current user's role can include lists not created by the current user.

The lists returned can include private lists with the same name, but created by different users, and role specific lists with the same name, but for different roles.

The following data is returned in apData:List:

- **RecordType.** The record type.
- **ListName.** The name of the list in the language of the currently logged-in user.
- **ListType.** The type of list; possible values are: Public, Private, Role Specific, System.
- **CreatedByAlias.** The alias of the user who created the list.
- **ModifiedByAlias.** The alias of the user who last modified the list.
- **RoleName.** For role specific lists only, the role to which the list is applicable.
- **CreatedById.** The identifier of the user who created the list.
- **ModifiedById.** The identifier of the user who last modified the list.
- **CreatedDate.** The date and time the list was created.
- **ModifiedDate.** The date and time the list was last modified.

- **Description.** A description of the list.
- **SearchIn.** The record set from which records are returned, for example, All records I can see, All records I own.
- **CaseInsensitiveFlag.** Whether fields are case sensitive in searches.
- **LanguageCode.** For system lists, the language code for the currently logged in user. For other lists, the language code in which the list was created.
- **ListOfFilterInformation.** The filter information for the list, contained in child elements of the <FilterInfo> element:
  - **Conjunction.** The operator And or Or.
  - **FieldName.** The system name of the field.
  - **Condition.** The condition for the filter.
  - **FilterValue.** The filter value.
  - **FilterValueAssocId.** The association ID (row ID) for a filter value for association control fields such as Contact Full Name, Account Name, Owner Alias, and so on, where the condition is Equal to.
- **ListOfSelectedFields.** The fields selected for display in search results contained in child elements of the <SelectedFieldInfo> element:
  - **SelectedField.** The system name of the field.
  - **SortOrder.** The sort order of a field used for sorting of search results: ASC for ascending or DESC for descending.
  - **SortPosition.** The position of a field within the list of fields used for sorting.

For more information about working with lists, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 132 describes the arguments taken by the ListRead method.

Table 132. Arguments Taken by the ListRead Method

Name	Description	Required	Default	I/O
apQuery:List	The list query criteria.	Yes	Not applicable	Input
apData:ListOfLists	The extracted list configuration data.	Not applicable	Not applicable	Output

## Return Value of the Call

The list data is returned.

## ListReadAll

Extracts all of the list configuration data for a company.

### Usage

Use the ListReadAll method to read all of the list data for all objects. See [“ListRead” on page 309](#) for information about the privileges required to access the various types of list.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including system list data. If <IncludeAll> specifies false, or is not included in the request, the method returns all configuration data except system list data.

See [“ListRead” on page 309](#) for information about the list data specified in apData:List.

### Arguments

[Table 133](#) describes the arguments taken by the ListReadAll method.

Table 133. Arguments Taken by the ListReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all list configuration data including system list data.	No	false	Input
apData:ListOfLists	The extracted list data.	Not applicable	Not applicable	Output

### Return Value of the Call

The list data is returned.

## ListAccessAndOrderRead

Extracts all of the list access and order configuration data for each role for an object.

### Usage

Use the ListAccessAndOrderRead method to read the list access and display order data for an object.

The object is specified with the <ObjectName> child element of apQuery:ListAccessAndOrder.

See [“ListAccessAndOrderUpsert” on page 314](#) for information about the list access and display order data specified in apData:ListAccessAndOrder.



## Arguments

[Table 134](#) describes the arguments taken by the ListAccessAndOrderRead method.

Table 134. Arguments Taken by the ListAccessAndOrderRead Method

Name	Description	Required	Default	I/O
apQuery:ListAccessAndOrder	The list access query criteria.	Yes	Not applicable	Input
apData:ListOfListAccessAndOrder	The extracted list access and display order data.	Not applicable	Not applicable	Output

## Return Value of the Call

The list access data is returned.

## ListAccessAndOrderReadAll

Extracts all of the list access and display order configuration data for a company.

## Usage

Use the ListAccessAndOrderReadAll method to read all of the list access and display order data for all objects.

See [“ListAccessAndOrderUpsert” on page 314](#) for information about the list access and display order data specified in apData:ListAccessAndOrder.

## Arguments

[Table 135](#) describes the arguments taken by the ListAccessAndOrderReadAll method.

Table 135. Arguments Taken by the ListAccessAndOrderReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfListAccessAndOrder	The extracted list access and display order data.	Not applicable	Not applicable	Output

## Return Value of the Call

The list access and display order data is returned.

## ListAccessAndOrderUpsert

Updates existing list access and display order data or inserts new data.

### Usage

Use the ListAccessAndOrderUpsert method to insert or update list access and order data for each role for an object.

The apdata:ListAccessAndOrder argument contains the following elements containing list access and display order data:

- **ObjectName.** The name of the object (required).
- **RoleName.** The name of the role (required).
- **ModifiedBy.** The username and time and date for the user who last updated the List Order Layout (read-only).
- **ListOfListAccessAndOrderList.** Contains all of the lists available for a role. The data for a particular list is contained in the following child elements of <ListAccessAndOrderList>:
  - **ListName.** The name of the list.
  - **Order.** An integer value indicating the list box in the List Order Layout page in which the list appears and the display order, if it appears in the Show in Short List box:
    - **-2.** The list appears in the All Lists box.
    - **-1.** The list appears in the Lists Available for this Role box.
    - **2 through n.** The list appears in the Show in Short List box and the integer value indicates the order of the list within the box starting with the value 2 for the topmost list, and so on.

See *Oracle Migration Tool On Demand Guide* for reference tables about the mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

### Arguments

Table 136 describes the arguments taken by the ListAccessAndOrderUpsert method.

Table 136. Arguments Taken by the ListAccessAndOrderUpsert Method

Name	Description	Required	Default	I/O
apData:ListAccessAndOrder	The list access and display order 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.

## 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 316 for information about the page layout field data specified in apData:PageLayout.

### Arguments

[Table 137](#) describes the arguments taken by the PageLayoutFieldRead method.

Table 137. Arguments Taken by the PageLayoutFieldRead Method

Name	Description	Required	Default	I/O
apQuery:PageLayout	The page layout query criteria.	Yes	Not applicable	Input
apData:ListOfPageLayout	The extracted page layout field data.	Not applicable	Not applicable	Output

### Return Value of the Call

The page layout fields are returned.

## PageLayoutFieldReadAll

Extracts all of the field configuration data for detail pages for a company.

### Usage

Use the PageLayoutFieldReadAll method to read all of the field data for detail pages.

If the input argument <CurrentUserOnly> specifies true, all of the page layout field data for the current user only is returned.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 27, and specified values up to and including 27 return configuration data applicable to Oracle CRM On Demand Release 27.

If the value specified is greater than 27, then all of the configuration data up to the release specified is returned. For example, if a value of 28 is specified, then all of the configuration data in Release 27 plus the newer data up to Release 28 is returned.

See [“PageLayoutFieldUpsert” on page 316](#) for information about the page layout field data specified in apData:PageLayout.

## Arguments

[Table 138](#) describes the arguments taken by the PageLayoutFieldReadAll method.

Table 138. 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
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	27	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.
  - **AlwaysOnLayout.** Whether the field is always on the page layout. A value of true indicates that the field cannot be removed from the page layout by the user.
  - **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 mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

## Arguments

Table 139 describes the arguments taken by the `PageLayoutFieldUpsert` method.

Table 139. Arguments Taken by the `PageLayoutFieldUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:PageLayout</code>	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 319](#) for information about the page layout related information specified in `apData:PageLayoutRelatedInformation`.

## Arguments

[Table 140](#) describes the arguments taken by the `PageLayoutRelatedInformationRead` method.

Table 140. 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 319 for information about the page layout related information specified in apData:PageLayoutRelatedInformation.

## Arguments

Table 141 describes the arguments taken by the PageLayoutRelatedInformationReadAll method.

Table 141. Arguments Taken by the PageLayoutRelatedInformationReadAll Method

Name	Description	Required	Default	I/O
CurrentUserOnly	If true, extracts data for the current user only	No	false	Input
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfPageLayoutRelatedInformation	The extracted page layout related information.	Not applicable	Not applicable	Output

## Return Value of the Call

The page layout related information is returned.

## PageLayoutRelatedInformationUpsert

Updates existing page layout related information or inserts new page layout related information.

### Usage

Use the PageLayoutRelatedInformationUpsert method to insert or update related information for a detail page layout.

The apdata:PageLayoutRelatedInformation argument contains the following elements containing related information for a page layout:

- **ObjectName.** The name of the object.
- **LayoutName.** The name of the page layout.
- **ListOfPageRelatedInformation.** Contains all of the related information for the page layout. The data for a particular related information section is contained in the following child elements of <PageRelatedInformation>:

- **Name.** A child object ID, equivalent to the name of a related information section. See *Oracle Migration Tool On Demand Guide* for reference tables about the mapping of child object IDs to UI display names for the child objects.
- **Availability.** An integer with the range of values -1 through 1, specifying in which list the related information appears in the page layout: Not Available, Available, or Displayed respectively.

For values of 1 (Displayed), the order of the <PageRelatedInformation> elements determines the order in which the related information sections are listed in the layout in the UI.

- **RelatedInformationLayout.** The name of the corresponding related information layout.

See *Oracle Migration Tool On Demand Guide* for reference tables about the mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

## Arguments

Table 142 describes the arguments taken by the PageLayoutRelatedInformationUpsert method.

Table 142. Arguments Taken by the PageLayoutRelatedInformationUpsert Method

Name	Description	Required	Default	I/O
apData:PageLayoutRelatedInformation	The page layout related information to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

## Return Value of the Call

The status is returned.

## 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 321](#) for information about the page layout section data specified in `apData:PageLayoutFormSection`.

## Arguments

[Table 143](#) describes the arguments taken by the `PageLayoutSectionReadAll` method.

Table 143. 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 is 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 mappings of three-letter language codes to languages.

## Arguments

Table 144 describes the arguments taken by the PageLayoutSectionUpsert method.

Table 144. Arguments Taken by the PageLayoutSectionUpsert Method

Name	Description	Required	Default	I/O
apData:PageLayoutForm Section	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, then the method returns configuration data including preconfigured and customized data. However, the data returned does not include any pre-configured read-only picklist data. If <IncludeAll> specifies false, or is not included in the request, then the method returns customized configuration data only.

You can use the <Disabled> child element of <PickListValue> to filter for active and inactive (disabled) picklist values for a particular picklist. To filter, you must specify <ObjectName> and <Name> to identify the picklist, and a value for <Disabled>. If you specify a value of true for <Disabled>, all inactive picklist values are returned, and if you specify false, all active picklist values are returned. If you do not specify a value for <Disabled>, both active and inactive picklist values are returned.

See [“PicklistUpsert” on page 324](#) for information about the picklist data specified in apData:PicklistSet.

## Arguments

Table 145 describes the arguments taken by the PicklistRead method.

Table 145. 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 except read-only picklists.

If the input argument <IncludeAll> specifies true, the method returns all configuration data, including preconfigured and customized data. However, the data returned does not include any preconfigured read-only picklist data. If <IncludeAll> specifies false, or is not included in the request, the method returns customized configuration data only.

See “PicklistUpsert” on page 324 for information about the picklist data specified in apData:PicklistSet.

## Arguments

Table 146 describes the arguments taken by the PicklistReadAll method.

Table 146. 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

Table 146. Arguments Taken by the PicklistReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	29	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 for each picklist:

- **Name.** The system name of the picklist field (required). When performing an insert operation, this name is used as the display name. You can use the FieldManagementRead method to obtain the system name of the picklist field.
- **WebSvcTag.** The Web Services v2.0 integration tag name for the picklist field. This element is read-only and is ignored in upsert operations.
- **ListOfPicklistValues.** The values for the picklist as contained in child elements of <PickListValue>:
  - **ValueId.** An identifier assigned to the picklist value. This element maps to the Id field in the Edit Picklist window for the picklist field in the UI. It is the language-independent name for the picklist value.
  - **Disabled.** Whether the picklist value is disabled.
  - **ListOfValueTranslations.** The picklist values in the languages activated for your company, contained in child elements of <ValueTranslation>:
    - **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 147](#) describes the arguments taken by the PicklistUpsert method.

Table 147. 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.

## PicklistValueGroupRead

Extracts picklist value group configuration data.

### Usage

Use the PicklistValueGroupRead method to extract picklist value group configuration data.

The picklist value group is specified with the <PicklistValueGroupName> or <PicklistValueGroupId> child element of apQuery:PicklistValueGroup so that you can query by both picklist value group name and row ID.

See [“PicklistValueGroupUpsert” on page 327](#) for information about the picklist value group data specified in apData:PicklistValueGroupSet.

## Arguments

Table 148 describes the arguments taken by the PicklistValueGroupRead method.

Table 148. Arguments Taken by the PicklistValueGroupRead Method

Name	Description	Required	Default	I/O
apQuery:ListOfPicklistValueGroup	The picklist value group query criteria.	Yes	Not applicable	Input
apData:ListOfPicklistValueGroupSet	The extracted picklist value group data.	Not applicable	Not applicable	Output

## Return Value of the Call

The picklist value group is returned.

## PicklistValueGroupReadAll

Extracts all of the picklist value group configuration data for a company.

## Usage

Use the PicklistValueGroupReadAll method to read all of the picklist value groups.

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 “[PicklistValueGroupUpsert](#)” on page 327 for information about the picklist value group data specified in apData:PicklistValueGroupSet.

## Arguments

Table 149 describes the arguments taken by the PicklistValueGroupReadAll method.

Table 149. Arguments Taken by the PicklistValueGroupReadAll Method

Name	Description	Required	Default	I/O
IncludeAll	If true, extracts all configuration data and not just customized configuration data.	No	false	Input
apData:ListOfPicklistValueGroupSet	The extracted data for all picklist value groups.	Not applicable	Not applicable	Output

## Return Value of the Call

The picklist value groups are returned.

## PicklistValueGroupUpsert

Updates an existing set of picklist value groups or inserts a new set of picklist value groups.

### Usage

Use the `PicklistValueGroupUpsert` method to insert a set of picklist value groups or update an existing set of picklist value groups.

The `apdata:PicklistValueGroupSet` argument contains the following elements containing picklist value group data:

- **PicklistValueGroupName.** The name of the picklist value group.
- **PicklistValueGroupId.** The row ID of the picklist value group.
- **Description.** A description of the picklist value group.
- **ListOfPicklistTypeSet.** The picklists that are mapped to the picklist value group:
  - **ObjectName.** The name of the object containing the picklist.
  - **FieldName.** The field name of the picklist.
- **ListOfLicNameSet.** The language-independent picklist values mapped to the picklist value group:
  - **LicName.** The language-independent picklist value.

### Arguments

[Table 150](#) describes the arguments taken by the `PicklistValueGroupUpsert` method.

Table 150. Arguments Taken by the `PicklistValueGroupUpsert` Method

Name	Description	Required	Default	I/O
<code>apData:ListOfPicklistValueGroupSet</code>	The picklist value group 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.

## ProcessAdministrationRead

Extracts process administration configuration data.

### Usage

Use the ProcessAdministrationRead method to extract process administration configuration data. The Process Administration privilege is required.

**NOTE:** The ProcessAdministrationRead method does not read the field setup and related access control configuration data for transition states.

You can query process administration data using the following child elements of apQuery:Process:

- <RecordType>
- <ProcessName>
- <RecordType> + <ProcessName>

See “[ProcessAdministrationUpsert](#)” on page 329 for information about the process administration data specified in apData:Process.

### Arguments

Table 151 describes the arguments taken by the ProcessAdministrationRead method.

Table 151. Arguments Taken by the ProcessAdministrationRead Method

Name	Description	Required	Default	I/O
apQuery:Process	The process administration query criteria.	Yes	Not applicable	Input
apData:ListOfProcess	The extracted process administration data.	Not applicable	Not applicable	Output

### Return Value of the Call

The process administration data is returned.

## ProcessAdministrationReadAll

Extracts all of the process administration configuration data for a company.



## Usage

Use the `ProcessAdministrationReadAll` method to read all of the process administration data. The Process Administration privilege is required.

**NOTE:** The `ProcessAdministrationReadAll` method does not read the field setup and related access control configuration data for transition states.

See “[ProcessAdministrationUpsert](#)” on page 329 for information about the process administration data specified in `apData:Process`.

## Arguments

Table 152 describes the arguments taken by the `ProcessAdministrationReadAll` method.

Table 152. Arguments Taken by the `ProcessAdministrationReadAll` Method

Name	Description	Required	Default	I/O
<code>apData:ListOfProcess</code>	The extracted process administration data.	Not applicable	Not applicable	Output

## Return Value of the Call

The process administration data is returned.

## ProcessAdministrationUpsert

Updates existing process administration data or inserts new process administration data.

## Usage

Use the `ProcessAdministrationUpsert` method to insert process administration data or update existing process administration data. The Process Administration privilege is required.

**NOTE:** The `ProcessAdministrationUpsert` method does not insert or update the field setup and related access control configuration data for transition states.

The `apdata:Process` argument contains the following elements containing process administration data:

- **RecordType.** The record type (required).
- **ProcessName.** The name of the process (required).
- **PrimaryField.** The field that is used as a filter to restrict the process to certain records of the record type (read-only). Only the Activity record type supports the use of a primary field.
- **TransitionField.** The field in the record type that determines the transition of records from one transition state to another (read-only). The Status field is used as the transition field for all supported record types.

- **Description.** A description of the process.
- **Order.** The order of the process within the record type (read-only).
- **Default.** Whether the process is the default process within the record type (read-only).
- **ListOfPrimaryValues.** The primary values for the process, which are contained in <Value> child elements of <PrimaryValues>.
- **ListOfTransitionStates.** The transition states for the process, contained in the following child elements of <TransitionState>.
  - **State.** The transition state (required). Possible values depend on the record type. For upsert operations, the value in the <State> element must match an existing picklist value in the Status field for the record type.
  - **Category.** The sales category with which the transition state is associated.
  - **Description.** A description of the transition state.
  - **Created.** The user who created the transition state data, with the date and time of creation (read-only).
  - **Modified.** The user who modified the transition state data, with the date and time of modification (read-only).
  - **Condition.** An expression that determines whether a record can move to the transition state. You cannot use any calculated fields in the expression.
  - **ErrorMessage.** The error message that appears if the condition set for the transition state is not met when a user tries to move a record of the record type specified in the process to the transition state.
  - **DisableUpdate.** Whether records can be updated (false) or not (true) when they are in the transition state.
  - **DisableDeleteOrRemove.** Whether records can be deleted (false) or not (true) when they are in the transition state.

For upsert operations <RecordType> is a user key. For transition states, the user key is <State>.

For more information about process administration, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 153 describes the arguments taken by the ProcessAdministrationUpsert method.

Table 153. Arguments Taken by the ProcessAdministrationUpsert Method

Name	Description	Required	Default	I/O
apData:Process	The process administration 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.

The <Version> child element of apQuery:RelatedInformationLayout specifies the release of Oracle CRM On Demand for which configuration data is extracted. The default value is 31, so to extract Release 32 or later functionality, you must specify a higher value than 31.

See [“RelatedInformationLayoutUpsert” on page 333](#) for information about the related information layout data specified in apData:RelatedInformationLayout.

### Arguments

[Table 154](#) describes the arguments taken by the RelatedInformationLayoutRead method.

Table 154. 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.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is extracted. The default value is 31, so to extract Release 32 or later functionality, you must specify a higher value than 31.

See [“RelatedInformationLayoutUpsert” on page 333](#) for information about the related information data specified in apData:RelatedInformationLayout.

### Arguments

[Table 155](#) describes the arguments taken by the RelatedInformationLayoutReadAll method.

Table 155. 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
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	31	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.
- **SortField.** (For some Opportunity child objects only.) The field used for sorting records, which can be one of the following values:
  - Primary Revenue Close Date. Corresponds to the Close Date field in the UI.
  - Created. Corresponds to the Created: Date field in the UI.
  - Name. Corresponds to the Opportunity Name field in the UI.
  - Updated. Corresponds to the Modified: Date field in the UI.
- **SortDir.** (For some Opportunity child objects only.) The sort order, which can be Ascending or Descending.
 

**NOTE:** Child opportunities can be sorted for the following objects: Account, Campaign, Contact, Custom Object 01 through Custom Object 03, Dealer, Objective, Opportunity child, Partner Program.
- **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 0 through 19, 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 156 describes the arguments taken by the RelatedInformationLayoutUpsert method.

Table 156. 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.

## ReportFolderRead

Extracts report and dashboard folder configuration data.

### Usage

Use the ReportFolderRead method to read the details of a specified folder and its role associations. The following privileges are required:

- Manage Roles and Access.
- To read role associations for V2 report folders and dashboard folders: Manage Content.
- To read role associations for report folders: Manage Custom Reports.
- To read role associations for dashboard folders: Manage Dashboards.
- To read role associations for the V3 migrated company wide shared folders: Access Migrated Company Wide Shared Folder.

The <Path> child element of apQuery:ReportFolder is required to query folders containing reports or dashboards. You can query using only the <Path> element, but you can also include the <Catalog>, <FolderName>, and <ParentFolder> child elements.

If <Catalog> is not specified in the request, and the path exists in both V2 and V3 catalogs, then results for both V2 and V3 catalogs are returned.

For the folders in the Company Wide Shared Folder for both the V2 catalog and V3 catalog, the root folder is /shared/Company Analyses, and for the folders in the Migrated Company Wide Shared Folder, the root folder is /shared/Migrated Company Analyses.

You can query the root folder with the <Path> element as in the following example:

```
<quer: Path>/shared/Company Analyses</quer: Path>
```

To query for a specific subfolder, you use the <Path> element as in the following example:

```
<quer: Path>/shared/Company Analyses/Acti vi ty Book Vi si bi li ty</quer: Path>.
```

where /shared/Company Analyses/Acti vi ty Book Vi si bi li ty is the path to that particular folder.

**NOTE:** Depending on how your company instance is set up, your company might not have access to the V3 catalog.

See [“ReportFolderUpsert” on page 336](#) for information about the folder data specified in apData:ReportFolder.

## Arguments

[Table 157](#) describes the arguments taken by the ReportFolderRead method.

Table 157. Arguments Taken by the ReportFolderRead Method

Name	Description	Required	Default	I/O
apQuery:ReportFolder	The folder query criteria.	Yes	Not applicable	Input
apData:ListOfReportFolder	The extracted folder data.	Not applicable	Not applicable	Output

## Return Value of the Call

The folder configuration data is returned.

## ReportFolderReadAll

Extracts all of the report and dashboard folder configuration data for a company.

## Usage

Use the ReportFolderReadAll method to read the details of all folders and their role associations. The following privileges are required:

- Manage Roles and Access
- To read role associations for V2 report folders and dashboard folders: Manage Content.
- To read role associations for report folders: Manage Custom Reports.
- To read role associations for dashboard folders: Manage Dashboards.
- To read role associations for the V3 migrated company wide shared folders: Access Migrated Company Wide Shared Folder.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 34, and specified values up to and including 34 return configuration data applicable to Oracle CRM On Demand Release 34.

ReportFolderReadAll returns folders, subfolders, and associated roles of all reports and dashboard folders from V2 and V3 catalogs namely:

- V2 Company Wide Shared Folder
- V3 Company Wide Shared Folder
- V3 Migrated Company Wide Shared Folder (only if your user role includes the Access Migrated Company Wide Shared Folder privilege).

**NOTE:** Depending on how your company instance is set up, your company might not have access to the V3 catalog.

See [“ReportFolderUpsert” on page 336](#) for information about the report folder data specified in apData:ReportFolder.

## Arguments

Table 158 describes the arguments taken by the ReportFolderReadAll method.

Table 158. Arguments Taken by the ReportFolderReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned	No	34	Input
apData:ListOfReportFolder	The extracted data for all folders.	Not applicable	Not applicable	Output

## Return Value of the Call

The folder configuration data is returned.

## ReportFolderUpsert

Updates or inserts role and access level information for a shared folder.

## Usage

Use the ReportFolderUpsert method to insert or update role and access level information for a shared folder. The following privileges are required:

- Manage Roles and Access



- To update and insert role associations for V2 report folders and dashboard folders: Manage Content.
- To update and insert role associations for report folders: Manage Custom Reports.
- To update and role associations for dashboard folders: Manage Dashboards.
- To update and insert role associations for the V3 migrated company wide shared folders: Access Migrated Company Wide Shared Folder.

ReportFolderUpsert can insert or update only the role and access level data for a report folder. It is not possible to insert or update any other report folder data. For upsert requests, <Path> is required.

The user keys for upsert operations are:

<Path>

<Path> + <Catalog>

If the <Catalog> element is not specified, and if multiple folders are found that match the value in the <Path> element, then an error is returned.

The root folder is read-only, so it is not possible to perform upserts for the root folder.

The apdata:ReportFolder argument contains the following elements containing report folder data:

- **Catalog.** The catalog version of the folder. Possible values are: V2 and V3.  
**NOTE:** Depending on how your company instance is set up, your company might not have access to the V3 catalog.
- **Path.** The path of the folder (required).
- **FolderName.** The name of the folder.
- **ParentFolder.** The parent folder. If the folder is created directly in the company-wide shared folder, then the name of the parent folder is Company Analyses.
- **Description.** A description of the folder.
- **ListOfSubFolders.** The subfolders:
  - **Catalog.** The catalog version of the folder. Possible values are: V2 and V3.
  - **Path.** The path of the folder.
  - **FolderName.** The name of the subfolder.
  - **Description.** A description of the subfolder.
- **ListOfAssociatedRoles.** The role data:
  - **RoleName.** The name of the role.
  - **AccessLevel.** The access level: Read or Change/Delete.

## Arguments

[Table 159](#) describes the arguments taken by the ReportFolderUpsert method.

Table 159. Arguments Taken by the ReportFolderUpsert Method

Name	Description	Required	Default	I/O
apData:ReportFolder	The report or dashboard folder data to be inserted or updated.	Yes	Not applicable	Input
Status	The status of the call.	Not applicable	Not applicable	Output

## Return Value of the Call

The status is returned.

## RoleRead

Extracts role configuration data.

## Usage

Use the RoleRead method to extract role configuration data. The Manage Roles and Access privilege is required.

The role is specified with the <RoleName> child element of apQuery:Role.

See ["RoleUpsert" on page 339](#) for information about the role data specified in apData:Role.

## Arguments

[Table 160](#) describes the arguments taken by the RoleRead method.

Table 160. 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. The Manage Roles and Access privilege is required.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 33, and specified values up to and including 33 return configuration data applicable to Oracle CRM On Demand Release 33.

If the value specified is greater than 33, then all of the configuration data up to the release specified is returned. For example, if a value of 34 is specified, then all of the configuration data in Release 33 plus the newer data up to Release 34 is returned.

See [“RoleUpsert” on page 339](#) for information about the role data specified in apData:Role.

### Arguments

[Table 161](#) describes the arguments taken by the RoleReadAll method.

Table 161. Arguments Taken by the RoleReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	33	Input
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 Manage Roles and Access privilege is required. The Manage Company privilege is required to update the <ExpirePasswordIn> element.

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.
- **ModernThemeName**. The tablet 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.
- **ExpirePasswordIn**. The length of time for which the password of users assigned this role is valid. Possible values are: 30 days, 60 days, 90 days, 180 days, One Year, Never expires. If no value is specified, then the password expiry period that is defined for the company applies to users who have this role.
- **ListOfRoleTranslation**. The role name in the languages activated for your company:
  - **LanguageCode**. The three-letter language code for the language.
  - **RoleName**. The role name in the language identified by the language code.
- **ListOfRecordTypeAccess**. The record type access data:
  - **RecordName**. The record type.
  - **HasAccess**. Whether users assigned this role have access to records of this type.
  - **CanCreate**. Whether users assigned this role can create records of this type.
  - **CanReadAll**. Whether users assigned this role can see all records of this type.
- **AccessProfile**. The access profiles associated with the role:
  - **DefaultAccessProfile**. The default access profile.
  - **OwnerAccessProfile**. The owner access profile.
- **ListOfPrivilege**. The privileges associated with the role:
  - **PrivilegeName**. A privilege ID, equivalent to the name of a privilege in the Oracle CRM On Demand application. See *Oracle Migration Tool On Demand Guide* for reference tables about the mapping of privilege IDs to UI display names for the privileges.
  - **Enabled**. Whether the privilege is assigned to the role.
- **TabAccessandOrder**. The tabs available and selected for the role:
  - **ListOfAvailableTab**. The available tabs:
    - **AvailableTab**. The name of the tab.
  - **ListOfSelectedTab**. The selected tabs:

- ❑ **TabName.** The name of the tab.
- ❑ **Order.** The position in which the tab name appears in the Selected Tabs list. A value of 0 (zero) corresponds to the first tab in the Selected Tabs list.
- **ListOfPageLayoutAssignment.** The page layout assignment for each record type:
  - **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 162 describes the arguments taken by the RoleUpsert method.

Table 162. 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 343](#) for information about the sales assessment template data specified in apData:SalesAssessmentTemplate.

## Arguments

[Table 163](#) describes the arguments taken by the SalesAssessmentTemplateRead method.

Table 163. Arguments Taken by the SalesAssessmentTemplateRead Method

Name	Description	Required	Default	I/O
apQuery:SalesAssessmentTemplate	The sales assessment template query criteria.	Yes	Not applicable	Input
apData:ListOfSalesAssessmentTemplate	The extracted sales assessment template data.	Not applicable	Not applicable	Output

## Return Value of the Call

The sales assessment template is returned.

## SalesAssessmentTemplateReadAll

Extracts all of the sales assessment template configuration data for a company.

## Usage

Use the SalesAssessmentTemplateReadAll method to read all of the sales assessment templates.

See [“SalesAssessmentTemplateUpsert” on page 343](#) 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 164 describes the arguments taken by the SalesAssessmentTemplateReadAll method.

Table 164. 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 165 describes the arguments taken by the SalesAssessmentTemplateUpsert method.

Table 165. 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.

## SalesCategoryRead

Extracts sales stage category configuration data.

### Usage

Use the SalesCategoryRead method to extract sales stage category configuration data matching the input criteria. The Manage Sales Stage Definition privilege is required.

The sales category is specified with the Order child element of apQuery:SalesCategorySet.

See [“SalesCategoryUpsert” on page 346](#) for information about the sales stage category data specified in apData:SalesCategory.

### Arguments

[Table 166](#) describes the arguments taken by the SalesCategoryRead method.

Table 166. Arguments Taken by the SalesCategoryRead Method

Name	Description	Required	Default	I/O
apQuery:SalesCategorySet	The sales stage category query criteria.	Yes	Not applicable	Input
apData:ListOfSalesCategorySet	The extracted sales stage category data.	Not applicable	Not applicable	Output

## Return Value of the Call

The sales stage category is returned.

## SalesCategoryReadAll

Extracts all of the sales stage category configuration data for a company.

### Usage

Use the SalesCategoryReadAll method to read all of the sales stage categories. The Manage Sales Stage Definition privilege is required.

See [“SalesCategoryUpsert” on page 346](#) for information about the sales stage category data specified in apData:SalesCategorySet.

## Arguments

Table 167 describes the arguments taken by the SalesCategoryReadAll method.

Table 167. Arguments Taken by the SalesCategoryReadAll Method

Name	Description	Required	Default	I/O
apData:ListOfSalesCategory Set	The extracted data for all sales stage categories.	Not applicable	Not applicable	Output

## Return Value of the Call

The sales stage categories are returned.

## SalesCategoryUpsert

Updates an existing sales stage category or inserts a new sales stage category.

### Usage

Use the SalesCategoryUpsert method to insert a sales stage category or update an existing sales stage category. The Manage Sales Stage Definition privilege is required.

The apdata:SalesCategorySet argument contains the following elements containing sales stage category data:

- **Order.** The sequence of the sales stage category.
- **StageCategory.** The name of the sales stage category.
- **Description.** A description for the sales stage category.
- **Mark for Translation.** Whether the sales stage category data is translated.
- **ListOfSalesCategoryTranslations.** The sales stage category data in the languages activated for your company:
  - **LanguageCode.** The three-letter language code for the language.
  - **StageCategory.** The name of the sales stage category in the language identified by the language code.
  - **Description.** The description of the sales stage category 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.

For more information about setting up sales stage categories, see *Oracle CRM On Demand Online Help*.

## Arguments

[Table 168](#) describes the arguments taken by the SalesCategoryUpsert method.

Table 168. Arguments Taken by the SalesCategoryUpsert Method

Name	Description	Required	Default	I/O
apData:SalesCategorySet	The sales stage category 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.

## SearchLayoutRead

Extracts all of the configuration data for a search layout for an object.

## Usage

Use the SearchLayoutRead method to read the search layout data for an object.

The object and layout are specified with the <ObjectName> and <LayoutName> child elements of <apQuery:SearchLayout>. Only fields included in the request are returned in the response.

See [“SearchLayoutUpsert” on page 348](#) for information about the search layout data specified in apData:ListOfSearchLayout.

## Arguments

[Table 169](#) describes the arguments taken by the SearchLayoutRead method.

Table 169. Arguments Taken by the SearchLayoutRead Method

Name	Description	Required	Default	I/O
apQuery:SearchLayout	The search layout query criteria.	Yes	Not applicable	Input
apData:ListOfSearchLayout	The extracted search layout data.	Not applicable	Not applicable	Output

## Return Value of the Call

The search layout is returned.

## SearchLayoutReadAll

Extracts all of the search layout configuration data for a company.

### Usage

Use the SearchLayoutReadAll method to read all of the search layout data.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 33, and specified values up to and including 33 return configuration data applicable to Oracle CRM On Demand Release 33.

If the value specified is greater than 33, then all of the configuration data up to the release specified is returned. For example, if a value of 34 is specified, then all of the configuration data in Release 33 plus the newer data up to Release 34 is returned.

See [“SearchLayoutUpsert” on page 348](#) for information about the search layout data specified in apData:ListOfSearchLayout.

### Arguments

[Table 170](#) describes the arguments taken by the SearchLayoutReadAll method.

Table 170. Arguments Taken by the SearchLayoutReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	33	Input
apData:ListOfSearchLayout	The extracted search layout data.	Not applicable	Not applicable	Output

### Return Value of the Call

The search layouts are returned.

## SearchLayoutUpsert

Updates existing search layout data or inserts new search layout data.

### Usage

Use the SearchLayoutUpsert method to insert or update search layout data.

The apdata:SearchLayout argument contains the following elements containing search layout data:

- **ObjectName.** The name of the object (required).

- **LayoutName.** The name of the search layout, which must be unique for an object (required).
- **LastUpdatedBy.** The date and time the layout was last updated.
- **Description.** A description of the search layout.
- **ListOfSpecifySearchFields.** Contains all of the fields specified for the search, corresponding to Step 2 in the Search Layout wizard in the UI. The field data is contained in the following child elements of <SearchLayoutField>:
  - **FieldName.** The system name of the field.
  - **SearchOptimization.** An integer value as follows (read-only):
    - -1. The field is not optimized for search and is always case sensitive (default text color in UI).
    - 1. The field is optimized for fast search and is always case sensitive (green in UI).
    - 2. The field is optimized for fast search and becomes case-insensitive when the Case Insensitive check box is selected in a search (blue in UI).
    - 3. The field is case-insensitive regardless of the Case insensitive check box and is not optimized for fast search (purple in UI).
  - **Section.** An integer identifying the section of the search layout in which the field appears:
    - 0. Not Available Search Fields.
    - 1. Targeted Search Fields.
    - 2. Additional Search Fields.
  - **Order.** The order of the field within the section, starting with 0 for the field at the top of the section. For the Not Available Search Fields section, where the fields are not ordered, all values for fields are -1.
- **ListOfDefineSearchResultsLayout.** Contains all of the fields specified for the search results, corresponding to Step 3 in the Search Layout wizard in the UI. The field data is contained in the following child elements of <SearchLayoutField>:
  - **FieldName.** The system name of the field.
  - **SearchOptimization.** An integer value as follows (read-only):
    - -1. The field is not optimized for search and is always case sensitive (default text color in UI).
    - 1. The field is optimized for fast search and is always case sensitive (green in UI).
    - 2. The field is optimized for fast search and becomes case-insensitive when the Case Insensitive check box is selected in a search (blue in UI).
    - 3. The field is case-insensitive regardless of the Case insensitive check box and is not optimized for fast search (purple in UI).
  - **Section.** An integer identifying the section of the search results in which the field appears.
    - 0. Not Available Search Results Fields.
    - 1. Lookup Window Search Results Fields.

### □ 2. Additional Search Results Fields.

- **Order.** The order of the field within the section, starting with 0 for the field at the top of the section. For the Not Available Search Fields section, where the fields are not ordered, all values for fields are -1.
- **EnableDivisionAsDefaultSearchField.** Whether the search layout for the Product record type lists products by division. Corresponds to the Configure Lookup Window with Division as the Default Search Field check box in the UI.
- **EnableDefaultFields.** Whether default search fields are used for the Lookup window. Corresponds to the Configure Lookup Window With Default Fields check box in the UI.
- **EnableDefaultSearchResults.** Whether the list of available records is displayed in the Lookup window for the object when it is opened. Corresponds to the Show Search Results When Lookup Window Opens check box in the UI.
- **EnableAutoResolve.** Whether smart associations are enabled. Corresponds to the Auto-Resolve Enabled check box in the UI.
- **EnableDefaultCtxSensitiveList.** Whether context sensitive lists appear before the other lists in the *record type* List field in the upper-left corner of the Lookup window. Corresponds to the Show Context-Sensitive List as Lookup Window Default check box in the UI.
- **EnableKeywordSearch.** Whether the results page for enhanced keyword search displays the set of fields defined in the search layout. Corresponds to the Configure Enhanced Keyword Search with Selected Search Results Fields check box in the UI.
- **EnableFirstTargetedFieldAsLookupDefault.** Whether the first field in the list of targeted search fields in the search layout is used as the default search field in the Lookup window. Corresponds to the First Targeted Search Field as Default Search in Lookups check box in the UI.
- **RestrictSearchToFieldsSelectedInRelatedInformationLayout.** Whether the search is restricted to fields in the related information layout for the object. In certain cases, this setting is not considered. For more information about these cases, see *Oracle CRM On Demand Online Help*.

See *Oracle CRM On Demand Online Help* for more information about searching and managing search layouts in Oracle CRM On Demand.

See *Oracle Migration Tool On Demand Guide* for reference tables about the mappings of object names to the display names for the objects in the Oracle CRM On Demand user interface.

## Arguments

[Table 171](#) describes the arguments taken by the SearchLayoutUpsert method.

Table 171. Arguments Taken by the SearchLayoutUpsert Method

Name	Description	Required	Default	I/O
apData: SearchLayout	The search 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.

## SSOTokenRead

Provides an SSO token to a Web services client.

## Usage

Use the SSOTokenRead method to request an SSO token that can be passed to the authentication servlet of Oracle CRM On Demand's Web application server to create an interactive session.

As an example of usage, the SSO Token service can provide a token that Web service applications use to create a session on the interactive server, and then request analytics reports or capture screen HTML. Requesting a token in this way is more secure than storing tokens in memory for entire sessions, and more convenient than prompting the user to re-enter credentials when making a UI request. For better performance, Web service applications should explicitly log off interactive sessions when they are done with them.

## Arguments

[Table 172](#) describes the arguments taken by the SSOTokenRead method.

Table 172. Arguments Taken by the SSOTokenRead Method

Name	Description	Required	Default	I/O
apQuery: SSOTokenQuery	The SSO token query.	Yes	Not applicable	Input
apData: SSOTokenData	The extracted SSO token.	Not applicable	Not applicable	Output

## Return Value of the Call

The SSO token 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`. If only `<WorkflowName>` is specified, all workflow action sets are returned.

If the value of `<Version>` is 31 or higher:

- You can read individual workflow actions by specifying a values for the `<Name>` element of `apQuery:WorkflowActionQuery`.
- You can read all the active workflow actions by specifying the `<Active>` element of `apQuery:WorkflowActionQuery`.
- You can use the child elements of `<WorkflowEmailTranslation>` to read the translations of the email subject and message body for the Send Email workflow action, for each language activated for your company, and if translations are available. For more information about the Send Email action, see *Oracle CRM On Demand Online Help*.
- You can use the `<BookExpr>` and `<ReplaceBookExpr>` child elements of `WorkflowAssignBookAction` to specify books assigned using the Assign a Book workflow action.
- The `<Order>` element is returned for each workflow action. For each workflow action, the `<Name>` and `<Active>` elements are also returned, otherwise only fields included in the request are returned.

See [“WorkflowActionSetUpsert” on page 355](#) for information about the workflow action data specified in `apQuery:WorkflowActionSet`.



## Arguments

Table 173 describes the arguments taken by the WorkflowActionSetRead method.

Table 173. 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 elements 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

Trigger Event	TriggerEvent LIC
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.
- **Version.** The release of Oracle CRM On Demand for which configuration data is extracted. The default value is 30, so to extract Release 31 or later functionality, you must specify a higher value than 30.

If the value of the <Version> element is 31 or higher:

- The child elements of <WorkflowEmailTranslation> containing translations of the email subject and message body for the Send Email workflow action, for each language activated for your company are returned, if available. For more information about the Send Email action, see *Oracle CRM On Demand Online Help*.
- The <BookExpr> and <ReplaceBookExpr> child elements of WorkflowAssignBookAction specifying books assigned using the Assign a Book workflow action are returned.
- The <Order> element is returned for each workflow action.

See [“WorkflowActionSetUpsert” on page 355](#) for information about the workflow action data specified in apQuery:WorkflowActionSet.

## Arguments

[Table 174](#) describes the arguments taken by the WorkflowActionSetReadForRuleSet method.

Table 174. 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.

**NOTE:** It is recommended that you use the `WorkflowActionSetInsertOrUpdate` method rather than the `WorkflowActionSetUpsert` method.

### Usage

Use the `WorkflowActionSetUpsert` method to update or insert workflow action configuration data. The Manage Workflow Rules privilege is required.

**CAUTION:** New workflow action sets are inserted. Existing workflow action sets are updated, but existing workflow action sets that are not in the request are deleted.

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 `<Version>` child element specifies the release of Oracle CRM On Demand for which configuration data is included. The default value is 30, so to include Release 31 or later functionality, you must specify a higher value than 30.

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.
- **Order.** The order number for the action within the set of actions.

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). The value of this element depends on the `<OwnerType>` element, as follows:
    - If `<OwnerType>` is set to U (Specific User), then the `<Owner>` element contains the name of the user.
    - If the `<OwnerType>` element is set to R (Relative User On Record), then the value of the `<Owner>` element is defined. For possible values, see *Oracle CRM On Demand Migration Tool Guide*.
  - **OwnerType.** The type of owner (required). `<OwnerType>` corresponds to the Owner field in the UI. Valid values are:
    - **R.** Corresponds to the Relative User on Record option in the UI.
    - **U.** Corresponds to the Specific User option in the UI.
  - **Type.** The activity type.
  - **SubType.** The activity subtype.
  - **DueDate.** The due date for the task (required).

- **Priority.** The activity priority (required). Pre-configured values are: 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.
  - **BookExpr.** An expression for a 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>.
  - **ReplaceBookExpr.** An expression for the book to be replaced.
  - **ApplyTo.** How the assignment option is to be applied (required): Manual Associations, Automatic Associations, or Both.

For the Assign a Book action, if both a book name and an expression are specified in the request, then only the book name is saved. Therefore, if you want to set up an Assign a Book action using an expression instead of the book name, you must make sure that only the expression field is populated.

- **WorkflowSyncOwnerPartnerAction.** The Sync Owner Partner action.
- **WorkflowSyncBookPartnerAction.** The Sync Book Partner action.
- **WorkflowSyncPartnerBooksAction.** The Sync Partner Books 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.
  - **SchemaVersion.** The Web service schema version that is to be used for the integration events. Web Services v2.0 is the default value.
- **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 type of sender email address (required). <FromType> corresponds to the From field in the UI. Valid values are:
    - **FROM\_EMAIL\_ADDRESS**. Corresponds to the Current User option in the UI.
    - **FROM\_DEFAULT**. Corresponds to the Default Email Address option in the UI.
    - **SPECIFIC\_EMAIL\_ADDRESS**. Corresponds to the Specific Email Address option in the UI.
  - **From**. The email address, if Specific Email Address is specified for <FromType>.
  - **ToType**. The type of the recipient for the email (required). <ToType> corresponds to the To field in the UI. Valid values are:
    - **R**. Corresponds to the Relative User on Record option in the UI.
    - **U**. Corresponds to the Specific User option in the UI.
    - **E**. Corresponds to the Specific Email Address option in the UI.
  - **To**. The recipient for the email, The value of this element depends on the value in the <ToType> element, as follows:
    - If <ToType> is set to U (Specific User), then the <To> element contains the User ID of the user.
    - If <ToType> is set to R (Relative User on Record), then the value of the <To> element is defined. For possible values, see *Oracle CRM On Demand Migration Tool Guide*.
    - If <ToType> is set to E (Specific Email Address), then the <To> element contains the expression defined for specific email addresses.

- **Subject.** The subject of the email (required).
- **EmailFormat.** The format of the email, which can be Plain Text or HTML.
- **MessageBody.** The message body (required). If you select HTML as the email format, you must enclose the HTML code in a CDATA section. Any unsupported elements are removed from the HTML. Refer to *Oracle CRM On Demand Online Help* for a list of the HTML elements that are allowed in the message body.
- **ListOfWorkflowEmailTranslations.** The email subject and message bodies in the languages activated for your company.
  - **LanguageCode.** The three-letter language code for the language.
  - **SubjectTranslation.** The subject of the email in the language identified by the language code.
  - **MessageBodyTranslation.** The message body in the language identified by the language code.

If you update the email subject and message body in the default company language through a Web services call, the Mark for Translation check box in the Workflow Action Edit page in the UI is automatically selected. The default company language is then used for the subject and message body for each translated language, unless you also provide translations for the language in the same Web services call.

- **WorkflowSubmitCallAction.** The Submit Call action.
- **WorkflowSubmitTransactionAction.** The Submit action for submitting sample transactions.

For more information about workflow rules and actions, see *Oracle CRM On Demand Online Help*.

The Submit Call and Submit action for submitting sample transactions are available only in Oracle CRM On Demand Life Sciences Edition. Note that there is no data associated with the <WorkflowSubmitCallAction> or <WorkflowSubmitTransactionAction> elements. For more information about the Submit Call action and how you can use it, see *Oracle CRM On Demand Disconnected Mobile Sales Administration Guide* on Oracle Technology Network. For more information about the Submit action for submitting sample transactions, see *Oracle CRM On Demand Online Help*.

The following workflow actions are available only in Oracle CRM On Demand for Partner Relationship Management:

- Sync Partner Books
- Sync Book Partner
- Sync Owner Partner

There is no data associated with the elements for these actions. 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.

## Arguments

Table 175 describes the arguments taken by the WorkflowActionSetUpsert method.

Table 175. 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.

## WorkflowActionSetInsertOrUpdate

Updates existing workflow action configuration data or inserts new data.

**NOTE:** It is recommended that you use the `WorkflowActionSetInsertOrUpdate` method rather than the `WorkflowActionSetUpsert` method.

## Usage

Use the `WorkflowActionSetInsertOrUpdate` method to update or insert workflow action configuration data.

New workflow actions are inserted and existing workflows actions are updated. Workflow actions that are not identified in the request are not affected. This is in contrast with the `WorkflowActionSetUpsert` method, which deletes workflow actions not identified in the request.

The `<WorkflowName>` and `<Order>` child elements of `apData:WorkflowActionSet` are used to identify workflow action records in a request.

If no workflow action is located using these criteria, a new workflow action is inserted with the configuration specified in the request.

See [“WorkflowRuleSetUpsert” on page 363](#) for information about the workflow action data specified in `apData:WorkflowActionSet`.

For more information about workflow actions, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 176 describes the arguments taken by the WorkflowActionSetInsertOrUpdate method.

Table 176. Arguments Taken by the WorkflowActionSetInsertOrUpdate 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.

You can delete workflow rule sets by specifying values for the <RecordType>, <TriggerEvent>, and <ParentRecordType> child elements of apQuery:WorkflowRuleSet. You can specify the following combinations of elements:

- <WorkflowName>
- <WorkflowName> and <RecordType>
- <WorkflowName> and <TriggerEvent>
- <WorkflowName> and <RecordType> and <TriggerEvent>
- <WorkflowName> and <RecordType> and <TriggerEvent> and <ParentRecordType>
- <RecordType> and <TriggerEvent>

You can delete an individual workflow rule by specifying a value for <WorkflowName>. Upon deletion of a workflow rule, the other workflow rules in the set are re-ordered accordingly.

When <WorkflowName>, <RecordType> and <TriggerEvent> are all specified for a delete request, only records that match all three values are deleted.

See [“WorkflowRuleSetUpsert” on page 363](#) for information about the workflow rule data specified in apQuery:WorkflowRuleSet.



## Arguments

Table 177 describes the arguments taken by the WorkflowRuleSetDelete method.

Table 177. 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.

You can read workflow rule sets by specifying the <RecordType>, <TriggerEvent>, and <ParentRecordType> child elements of apQuery:WorkflowRuleSet.

You can read an individual workflow rule by specifying a value for the <WorkflowName> element.

The <Order> element is returned for each workflow rule, otherwise only fields included in the request are returned.

See “WorkflowRuleSetUpsert” on page 363 for information about the workflow rule data specified in apData:WorkflowRuleSet.

## Arguments

Table 178 describes the arguments taken by the WorkflowRuleSetRead method.

Table 178. 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.

The input argument <Version> specifies the release of Oracle CRM On Demand for which configuration data is returned. The default value is 28, and specified values up to and including 28 return configuration data applicable to Oracle CRM On Demand Release 28.

If the value specified is greater than 28, then all of the configuration data up to the release specified is returned. For example, if a value of 29 is specified, then all of the configuration data in Release 28 plus the newer data up to Release 29 is returned. See [“WorkflowRuleSetUpsert” on page 363](#) for information about the workflow rule data specified in apData:WorkflowRuleSet.

The <Order> element is returned for each workflow rule.

### Arguments

[Table 179](#) describes the arguments taken by the WorkflowRuleSetReadAll method.

Table 179. Arguments Taken by the WorkflowRuleSetReadAll Method

Name	Description	Required	Default	I/O
Version	Specifies the release of Oracle CRM On Demand for which configuration data is returned.	No	28	Input
apData:ListOfWorkflowRuleSet	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 (to create workflow rules for the User record type, the Manage Workflow Rules for User privilege is also required). Additionally, the Workflow Cancel Save privilege is required if the Cancel save setting is to be updated or inserted.

**NOTE:** It is recommended that you use the `WorkflowRuleSetInsertOrUpdate` method rather than the `WorkflowRuleSetUpsert` method.

### Usage

Use the `WorkflowRuleSetUpsert` method to update or insert workflow rule configuration data. The Manage Workflow Rules privilege is required.

**CAUTION:** New workflow rules are inserted. Existing workflow rules are updated, but existing workflow rules that are not in the request are deleted.

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
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.
    - **Description.** A description of the workflow rule.
    - **WorkflowRuleCondition.** The condition for the workflow rule.
    - **CancelSave.** Whether Cancel Save processing is performed, if the condition is not met.
    - **Order.** The order number for the workflow rule. The order number determines the order in which Oracle CRM On Demand invokes a sequence of workflow rules that are based on the same record type and the same trigger event, and if applicable, on the same parent record type.
    - **DisplayMessage.** The message returned if the condition is met, and if cancel save processing is performed.
  - **ListOfWorkFlowTranslations.** Translations in the languages activated for your company of the display messages that are returned if the condition on a rule is not met and the save operation is canceled. Each <WorkflowTranslation> element has the following child elements:
    - **LanguageCode.** The three-letter language code for the language.
    - **DisplayMessageTranslation.** The translation in the language identified by the language code.
- NOTE:** The order of invoking of workflow rules is determined by the sequence of <WorkflowRule> elements.

For more information about workflow rules, see *Oracle CRM On Demand Online Help*.

## Arguments

Table 180 describes the arguments taken by the WorkflowRuleSetUpsert method.

Table 180. 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.

## WorkflowRuleSetInsertOrUpdate

Updates existing workflow rule configuration data or inserts new data. The Manage Workflow Rules privilege is required (to create workflow rules for the User record type, the Manage Workflow Rules for User privilege is also required). Additionally, the Workflow Cancel Save privilege is required if the Cancel Save setting is to be updated or inserted.

**NOTE:** It is recommended that you use the `WorkflowRuleSetInsertOrUpdate` method rather than the `WorkflowRuleSetUpsert` method.

## Usage

Use the `WorkflowRuleSetInsertOrUpdate` method to update or insert workflow rule configuration data.

New workflow rules are inserted and existing workflows rules are updated. Workflow rules that are not identified in the request are not affected. This is in contrast with the `WorkflowRuleSetUpsert` method, which deletes workflow rules not identified in the request.

Child elements of `apData:WorkflowRuleSet` provide user keys to identify workflow rule records in a request in the following order:

- 1 <RecordType> and <TriggerEvent> and <Order>
- 2 <WorkflowName>

The same order is followed for all the workflow child containers. If no workflow rule is located using these criteria, a new workflow is inserted with the configuration specified in the request.

See "[WorkflowRuleSetUpsert](#)" on page 363 for information about the workflow rule data specified in `apData:WorkflowRuleSet`.

For more information about workflow rules, see *Oracle CRM On Demand Online Help*.

## Arguments

[Table 181](#) describes the arguments taken by the `WorkflowRuleSetInsertOrUpdate` method.

Table 181. Arguments Taken by the `WorkflowRuleSetInsertOrUpdate` Method

Name	Description	Required	Default	I/O
<code>apData:WorkflowRuleSet</code>	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 182](#) lists the calls available with the Data Loader API.

Table 182. 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 183 on page 368](#).

The xsdLocal1: ImportOptions argument specifies the import options, which are similar to the options available in the Import Wizard in the Oracle CRM On Demand application:

- **CSVDelimiter.** The type of CSV delimiter used in the file: comma (,) or semi-colon (;).
- **DateTimeFormat.** The date and time format used in the CSV file: usa, can, eur, uk, other.
- **DuplicateCheckOption.** Whether the Row Id (rowid) or external ID (externalid) are used for duplicate checking.
- **ErrorLogLevel.** The level of error logging:

- all. All messages
- errors. Error messages only.
- errorswarnings. Error messages and warning messages only.

Typical customer import data is contained in a CSV file with the following format:

ColumnName1, ColumnName2, ...

Data1, Data2, ...

Data1, Data2, ...

...

The `xsdLocal1:ListOfFieldMappings` element specifies the mapping of CSV column names to field names in Oracle CRM On Demand. For example:

```
<ns1:ListOfFieldMappings>
```

```
  <ns1:FieldMapping>
```

```
    <ns1:CSVColumnName>
```

```
      Web Site
```

```
    </ns1:CSVColumnName>
```

```
    <ns1:FieldName>
```

```
      Home Page
```

```
    </ns1:FieldName>
```

```
  </ns1:FieldMapping>
```

```
</ns1:ListOfFieldMappings>
```

`CSVColumnHeaders` specifies the list of CSV column names, for example:

```
<ns0:CSVColumnHeaders>
```

```
  "Account Name", "External Unique Id", "Web Site", "Account Location"
```

```
</ns0:CSVColumnHeaders>
```

The mappings are validated and if everything is correct, the import request is created.

## Arguments

Table 183 describes the arguments taken by the BulkOpImportCreateRequest method.

Table 183. 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
EnableMailNotification	Enables or disables email notifications for import requests.  Accepted values are true or false.	No	true	Input
SendUserLogFile	Whether the user log file is attached to the email notification for import requests. If email notification is disabled, SendUserLogFile is ignored.  Accepted values are true or false.	No	false	Input
CSVRowCount	The number of data records in the data file to be imported.	Yes	Not applicable	Input



Table 183. Arguments Taken by the BulkOpImportCreateRequest Method

Name	Description	Required	Default	I/O
Operation	The operation for the current request, such as insert, update.	Yes	Not applicable	Input
ClientVersion	The current Data Loader client version number.	No	Not applicable	Input
RequestId	The ID of the request created.	Not applicable	Not applicable	Output
Status	The status of this Web service request.	Not applicable	Not applicable	Output
ErrorMessage	Any error message from this Web service request, or a message indicating success.	Not applicable	Not applicable	Output

### Return Value of the Call

The request ID, status, and any error message are returned.

## BulkOpImportSendData

Imports the data for a bulk import request.

### Usage

Use the BulkOpImportSendData method to import the data for a bulk import request.

The BulkOpImportCreateRequest call must be invoked before BulkOpImportSendData by the client.

...

The ListOfCSVData element specifies the row data from the CSV file. For example:

```
<ns1:ListOfCSVData>
  <ns1:CSVData>
    <ns1:Row>
      "High Quality Moving Company", "HQMC", "www.hi ghmovi ng. net", "San Jose"
    </ns1:Row>
  </ns1:CSVData>
</ns1:ListOfCSVData>
```

The CSV data is processed and the actual import performed.

## Arguments

Table 184 describes the arguments taken by the BulkOpImportSendData method.

Table 184. 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 185 describes the arguments taken by the BulkOpImportGetRequestDetail method.

Table 185. 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.



# 7

## Oracle CRM On Demand Objects Accessible Through Web Services

This chapter contains reference information about the objects accessible through the Web Services On Demand API. These objects correspond to record types and enable access to data stored within an instance of Oracle CRM On Demand. This chapter contains the following topics:

- [Reference Information About the Parent Objects on page 373](#)
- [Core Parent Objects \(Web Services v1.0\) on page 377](#)
- [Life Sciences Edition Parent Objects \(Web Services v1.0\) on page 491](#)
- [Financial Services Edition Parent Objects \(Web Services v1.0\) on page 495](#)
- [Automotive Edition Parent Objects \(Web Services v1.0\) on page 504](#)
- [Child Objects \(Web Services v1.0\) on page 510](#)
- [Core Parent Objects \(Web Services v2.0\) on page 518](#)
- [Partner Relationship Management Edition Objects \(Web Services v2.0\) on page 581](#)
- [Life Sciences Edition Parent Objects \(Web Services v2.0\) on page 613](#)
- [Financial Services Edition Parent Objects \(Web Services v2.0\) on page 670](#)
- [Automotive Edition Parent Objects \(Web Services v2.0\) on page 697](#)
- [Child Objects \(Web Services v2.0\) on page 701](#)

### Reference Information About the Parent Objects

The reference information about the parent objects (starting with [“Account” on page 378](#)) 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 [Appendix A, “Using Attachments with Web Services On Demand.”](#)
- The fields that are accessible for the objects:
  - The required and read-only fields, see [“Required and Read-Only Fields” on page 374](#)
  - The user keys, see [“Oracle CRM On Demand User Keys” on page 374](#)
  - The audit fields, see [“Audit Fields” on page 375](#)
  - The status key, see [“Oracle CRM On Demand Status Keys” on page 375](#)
  - The pick map fields, see [“Oracle CRM On Demand Pick Maps” on page 376](#)

- For Web Services v1.0 only, the filterable fields, see [“Filterable Fields” on page 377](#)
- The editable picklist fields

## Field Names in WSDL Files

The field names used in the WSDL files for Oracle CRM On Demand Web services and shown for each object in this chapter correspond to integration tags rather than display names.

For Web Services v2.0, the integration tag names are listed in the Integration Tag Web Services v2.0 column in the Fields list page in the UI for each record type.

## Required and Read-Only Fields

The required fields listed in this chapter are those that are required for insert or update operations. Also listed in the same table for each record type are fields that cannot be inserted or updated because they are read-only.

**NOTE:** Fields can be conditionally required. A conditionally required field is one that is not required by default, but which becomes required if the data that you enter in other fields on the record meets the criteria defined in an expression. You can configure a field to be conditionally required in the field setup page for each record type. For more information about conditionally-required fields, see *Oracle CRM On Demand Online Help*.

For a list of the fields for a record type, refer to the Fields list page in the UI for each record type.

## 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 377, and “[Core Parent Objects \(Web Services v2.0\)](#)” on page 518.

## 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 186](#).

Table 186. 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.
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 377](#).

For Web Services v2.0, a standard set of status key fields is returned for all parent and child objects:

- Id
- CreatedBy
- CreatedById
- CreatedDate
- ExternalSystemId
- 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 377](#) and [“Core Parent Objects \(Web Services v2.0\)” on page 518](#).



## 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 377](#). For the Web Services V2.0 API, all fields in parent objects are filterable.

## Core Parent Objects (Web Services v1.0)

This topic include the standard objects of Oracle CRM On Demand. Objects that are only available with industry-specific editions of the application are covered in the following topics:

- [“Life Sciences Edition Parent Objects \(Web Services v1.0\)” on page 491](#)
- [“Financial Services Edition Parent Objects \(Web Services v1.0\)” on page 495](#)
- [“Automotive Edition Parent Objects \(Web Services v1.0\)” on page 504](#)

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

- [“Account” on page 378](#)
- [“Activity” on page 397](#)
- [“Asset” on page 406](#)
- [“Book” on page 409](#)
- [“Campaign” on page 412](#)
- [“Contact” on page 421](#)
- [“Current User” on page 440](#)
- [“CustomObject1 - CustomObject3” on page 442](#)
- [“Lead” on page 451](#)
- [“Note” on page 458](#)
- [“Opportunity” on page 459](#)
- [“Product” on page 472](#)
- [“Product Category” on page 475](#)
- [“Service Request” on page 477](#)
- [“Solution” on page 481](#)
- [“Territory” on page 484](#)
- [“User” on page 486](#)
- [“User Group” on page 489](#)

## 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](#), [Shared Address](#), and [Team](#).

### Methods Called

[Table 187](#) details the methods called by the Account service.

Table 187. Methods Called by Account Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	AccountDelete
<a href="#">"DeleteChild" on page 92</a>	AccountDeleteChild
<a href="#">"Insert" on page 93</a>	AccountInsert
<a href="#">"InsertChild" on page 94</a>	AccountInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	AccountInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	AccountQueryPage

Table 187. Methods Called by Account Service

Method	Name as Defined in Service
<a href="#">"Update" on page 106</a>	AccountUpdate
<a href="#">"UpdateChild" on page 107</a>	AccountUpdateChild

## Fields

[Table 188](#) details the required and read-only fields for the account object.

Table 188. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Account (parent)	AccountName	Required
	AccountConcatField	Read-only
	<a href="#">Audit Fields</a>	Read-only
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTIME	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	DealerName	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 188. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	AccountId	Read-only
	Id	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only
Contact	Age	Read-only
	OwnerFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	LastActivityDate	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
CustomObject3	AccountCustomObject3CreatedById	Read-only
	AccountCustomObject3CreatedDate	Read-only
	AccountCustomObject3ModifiedById	Read-only
	AccountCustomObject3ModifiedDate	Read-only
	CustomObject3Id	Read-only

Table 188. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Lead	OwnerFullName	Read-only
	SalesRepFullName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	ReferredByFullName	Read-only
	FuriganaAccountName	Read-only
	FuriganaContactFirstName	Read-only
	FuriganaContactLastName	Read-only
	FirstNameShadow	Read-only
	LastNameShadow	Read-only
	CompanyNameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
Multiple Contact Roles	ContactRole	Required
	ContactId	Required
	<a href="#">Audit Fields</a>	Read-only
Opportunity	OwnerFullName	Read-only
	PrimaryRevenueExpectedValue	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	FuriganaAccountName	Read-only
	NameShadow	Read-only
	AssignmentStatus	Read-only
	LastAssignmentCompletionDate	Read-only
	LastAssignmentSubmissionDate	Read-only
	RelatedAccount	AccountRelationshipId
RelatedAccountId		Read-only

Table 188. Required and Read-Only Fields for the Account Object

Child Component	Field Name	Type
Revenue	RevenueId	Required
	PartNumber	Required
	Revenue	Required
	ContactFullName	Required
	<a href="#">Audit Fields</a>	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
SharedAddress	All fields	Read-only

Table 189 details the status key for the account object, and the child component on which this key resides.

Table 189. Status Key for the Account Object

Child Component	Field Name
Account (parent)	<a href="#">Audit Fields</a>
	ExternalSystemId
	AccountId
	IntegrationId
	LastUpdated

Table 189. Status Key for the Account Object

Child Component	Field Name
AccountNote	<a href="#">Audit Fields</a>
	ExternalSystemId
	AccountNoteId
	IntegrationId
AccountTeam	<a href="#">Audit Fields</a>
	AccountTeamId
Activity	<a href="#">Audit Fields</a>
	ExternalSystemId
	ActivityId
	IntegrationId
Asset	<a href="#">Audit Fields</a>
	ExternalSystemId
	AssetId
	IntegrationId
Attachment	<a href="#">Audit Fields</a>
	Id
	AccountId
	ModId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId
Competitor	<a href="#">Audit Fields</a>
	AccountCompetitorId
	CompetitorExternalId
	CompetitorId
	CompetitorIntegrationId
Contact	<a href="#">Audit Fields</a>
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId

Table 189. Status Key for the Account Object

Child Component	Field Name
CustomObject3	AccountCustomObject3CreatedById
	AccountCustomObject3CreatedDate
	AccountCustomObject3ModifiedById
	AccountCustomObject3ModifiedDate
Lead	<a href="#">Audit Fields</a>
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
Opportunity	<a href="#">Audit Fields</a>
	AccountId
	ExternalSystemId
	IntegrationId
	OpportunityId
RelatedAccount	<a href="#">Audit Fields</a>
	AccountRelationshipId
Revenue	<a href="#">Audit Fields</a>
	ExternalId
	IntegrationId
	RevenueId
ServiceRequest	<a href="#">Audit Fields</a>
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	ServiceRequestId



Table 189. Status Key for the Account Object

Child Component	Field Name
Partner	<a href="#">Audit Fields</a>
	AccountPartnerId
	PartnerExternalId
	PartnerId
	PartnerIntegrationId
	Updated
SharedAddress	<a href="#">Audit Fields</a>

Table 190 details the pick map fields for the account object and the child objects on which they reside.

Table 190. 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 190. 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 190. 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 190. 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 190. 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 190. 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 191 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 191. 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 191. Filterable Fields and User Key Field Combinations 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 191. 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 191. 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	
	ContactExternalId	ContactIntegrationId
	ContactIntegrationId	ContactExternalId
	ContactRole	
	ModifiedDate	
Opportunity	Owner	OpportunityId
	Revenue	Id
	CloseDate	IntegrationId
	Forecast	ExternalSystemId
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	Status	
	OpportunityId	
	Id	

Table 191. 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 191. 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 191. 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	
Shared Address	None	AddressId
		AddressExternalSystemId

Table 192 details the picklists available for the account object.

Table 192. 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

Table 192. Picklists Available for the Account Object

Child Component	Field Name
RelatedAccount	Relationship
	Status
Revenue	Type
	Status
	Frequency
SharedAddress	AddressType
	CurrencyCode

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 193](#) details the methods called by the Activity service.

Table 193. Methods Called by Activity Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	ActivityDelete
<a href="#">"DeleteChild" on page 92</a>	ActivityDeleteChild
<a href="#">"Insert" on page 93</a>	ActivityInsert
<a href="#">"InsertChild" on page 94</a>	ActivityInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	ActivityInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	ActivityQueryPage
<a href="#">"Update" on page 106</a>	ActivityUpdate
<a href="#">"UpdateChild" on page 107</a>	ActivityUpdateChild

## Fields

Table 194 details the required and read-only fields for the activity object.

Table 194. 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	
<a href="#">Audit Fields</a>	Read-only	
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	ActivityId	Read-only
	Id	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 194. 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
	<a href="#">Audit Fields</a>	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
	<a href="#">Audit Fields</a>	Read-only
ProductDetailed	ProductId	Required
	Indication	Required
	ProductDetailedId	Read-only
PromotionalItemDropped	ProductName	Required
	Quantity	Required
	ProductAllocationId	Required
	<a href="#">Audit Fields</a>	Read-only
SampleDropped	ProductId	Required
	Quantity	Required
	SampleDroppedId	Read-only
	<a href="#">Audit Fields</a>	Read-only
User	UserId	Read-only
	UserEmail	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only
	UserRole	Read-only
	<a href="#">Audit Fields</a>	Read-only



Table 195 details the status key for the activity object.

Table 195. Status Key for the Activity Object

Child Component	Field Name
Activity (parent)	<a href="#">Audit Fields</a>
	ActivityId
	ExternalSystemId
	IntegrationId
Attachment	<a href="#">Audit Fields</a>
	Id
	ActivityId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId
Contact	<a href="#">Audit Fields</a>
	ActivityContactId
	ContactIntegrationId
	ContactExternalSystemId
	ContactId
User	<a href="#">Audit Fields</a>
	UserId
	UserExternalSystemId
	UserIntegrationId
ProductDetailed	<a href="#">Audit Fields</a>
	ProductDetailedId
	ExternalId
SampleDropped	<a href="#">Audit Fields</a>
	SampleDroppedId
	ExternalId
Solution	<a href="#">Audit Fields</a>
	SolutionId
	ExternalId

Table 196 details the pick map fields for the activity object and the child objects on which they reside.

Table 196. 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 196. 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 197 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 197. 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 197. 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	
	ProductDetailedExternalSystemId	ProductDetailedExternalSystemId
	ProductId	ProductId
	ProductExternalSystemId	ProductExternalSystemId
	ModifiedDate	Name Indication
SampleDropped	SampleDroppedId	SampleDroppedId
	SampleDroppedExternalSystemId	SampleDroppedExternalSystemId
	ProductId	ProductId
	ProductExternalSystemId	ProductExternalSystemId
	ModifiedDate	Quantity ProductName

Table 197. 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 198 details the picklists available for the activity object.

Table 198. Picklists Available for the Activity Object

Field Name
AccountName
AccountLocation
AccountIntegrationId
AccountExternalSystemId
OpportunityName
ServiceRequestNumber
ServiceRequestIntegrationId
ServiceRequestExternalSystemId
DelegatedByExternalSystemId

Table 198. Picklists Available for the Activity Object

Field Name
PrimaryContactIntegrationId
PrimaryContactExternalSystemId
MedEdEventIntegrationId
MedEdEventExternalSystemId
FundRequestExternalId
LeadIntegrationId
LeadExternalSystemId
CampaignIntegrationId
CampaignExternalSystemId
ActivitySubtype
ResolutionCode
PublishInternal
Status
<b>NOTE:</b> The Status picklist is editable for users whose role includes the Customize Application - Edit Activity 'Status' Picklist privilege.
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 199](#) details the methods called by the Asset service.

Table 199. Methods Called by Asset Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	AssetDelete
<a href="#">"Insert" on page 93</a>	AssetInsert
<a href="#">"InsertOrUpdate" on page 95</a>	AssetInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	AssetQueryPage
<a href="#">"Update" on page 106</a>	AssetUpdate

## Fields

[Table 200](#) details the required and read-only fields for the asset object.

Table 200. 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
	<a href="#">Audit Fields</a>	Read-only

[Table 201](#) details the status key for the asset object.

Table 201. Status Key for the Asset Object

Child Component	Field Name
Asset (parent)	<a href="#">Audit Fields</a>
	AssetId
	IntegrationId
	ExternalSystemId

Table 202 details the pick map fields for the asset object.

Table 202. 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 203 provides a list of the filterable fields and a list of user key combinations for the asset object.

Table 203. 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 204 details the picklists available for the asset object.

Table 204. 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 205](#) details the methods called by the Book service.

Table 205. Methods Called by Book Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	BookDelete
<a href="#">"DeleteChild" on page 92</a>	BookDeleteChild
<a href="#">"Insert" on page 93</a>	BookInsert
<a href="#">"InsertChild" on page 94</a>	BookInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	BookInsertOrUpdate

Table 205. Methods Called by Book Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 96</a>	BookQueryPage
<a href="#">"Update" on page 106</a>	BookUpdate
<a href="#">"UpdateChild" on page 107</a>	BookUpdateChild

## Fields

[Table 206](#) details the required and read-only fields for the book object.

Table 206. Required and Read-Only Fields for the Book Object

Child Component	Field Name	Type
Book (parent)	BookName	Required
	BookId	Read-only
	<a href="#">Audit Fields</a>	Read-only
BookUser	UserAlias	Required
	AccessProfileName	Required
	<a href="#">Audit Fields</a>	Read-only
SubBook	BookName	Required
	<a href="#">Audit Fields</a>	Read-only

[Table 207](#) details the status key for the book object.

Table 207. Status Key for the Book Object

Child Component	Field Name
Book (parent)	<a href="#">Audit Fields</a>
	BookId
	BookName
BookUser	<a href="#">Audit Fields</a>
	BookId
SubBook	BookId
	UserId

Table 208 details the pick map fields for the book object.

Table 208. Pick Map Fields for the Book Object

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

Table 209 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 209. 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 210 details the picklists available for the book object.

Table 210. 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 211](#) details the methods called by the Campaign service.

Table 211. Methods Called by Campaign Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	CampaignDelete
<a href="#">"DeleteChild" on page 92</a>	CampaignDeleteChild
<a href="#">"Insert" on page 93</a>	CampaignInsert
<a href="#">"InsertChild" on page 94</a>	CampaignInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	CampaignInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	CampaignQueryPage
<a href="#">"Update" on page 106</a>	CampaignUpdate
<a href="#">"UpdateChild" on page 107</a>	CampaignUpdateChild

## Fields

Table 212 details the required and read-only fields for the campaign object.

Table 212. Required and Read-Only Fields for the Campaign Object

Child Object	Field Name	Type
Campaign	CampaignName	Required
	SourceCode	Required
	<a href="#">Audit Fields</a>	Read-only
	CreatedByFullName	Read-only
	LastUpdated	Read-only
Activity	AccountLocation	Read-only
	CreatedDetail	Read-only
	MEEventName	Read-only
	CreatedbyEmailAddress	Read-only
	ModifiedbyEmailAddress	Read-only
	CODInteractionTime	Read-only
	CODWrapUpTime	Read-only
	CODHandleTime	Read-only
	CODIVRTIME	Read-only
	CODQueueHoldTime	Read-only
	CODTotalHoldTime	Read-only
	DescriptionShadow	Read-only
	Duration	Read-only
	<a href="#">Audit Fields</a>	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	CampaignId	Read-only
	Id	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 212. 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
	<a href="#">Audit Fields</a>	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 213 details the status key for the campaign object.

Table 213. Status Key for the Campaign Object

Child Component	Field Name
Campaign (parent)	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
Activity	<a href="#">Audit Fields</a>
	CampaignId
	ExternalSystemId
	Id
	IntegrationId
Attachment	<a href="#">Audit Fields</a>
	Id
	CampaignId
CampaignNote	<a href="#">Audit Fields</a>
	CampaignId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Lead	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Opportunity	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 213. Status Key for the Campaign Object

Child Component	Field Name
Recipient	<a href="#">Audit Fields</a>
	CampaignContactId
	ContactExtrenalId
	ContactIntegrationId
	ContactId

[Table 214](#) details the pick map field for the campaign object.

Table 214. Pick Map Field for the Campaign Object

Child Component	Pick Map Field	Maps To
Campaign (parent)	Owner	OwnerId



Table 214. 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 214. 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 215 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 215. 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 215. Filterable Fields and User Key Fields on the Campaign Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Opportunity	Account	OpportunityName
	Owner	IntegrationId
	Revenue	ExternalSystemId
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
Recipient	ContactId	
	ModifiedDate	ContactIntegrationId
		ContactExternalSystemId
		CampaignContactId

Table 216 details the picklists available for the campaign object.

Table 216. 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](#), [Household](#), [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](#), [Shared Address](#), and [Team](#).

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

### Methods Called

[Table 217](#) details the methods called by the Contact service.

Table 217. Methods Called by Contact Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	ContactDelete
<a href="#">"DeleteChild" on page 92</a>	ContactDeleteChild
<a href="#">"Insert" on page 93</a>	ContactInsert
<a href="#">"InsertChild" on page 94</a>	ContactInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	ContactInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	ContactQueryPage
<a href="#">"Update" on page 106</a>	ContactUpdate
<a href="#">"UpdateChild" on page 107</a>	ContactUpdateChild

## Fields

Table 218 details the required and read-only fields for the contact object.

Table 218. 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
	<a href="#">Audit Fields</a>	Read-only
	Manager	Read-only
	PrimaryAddressId	Read-only
Account	AccountId	Read-only
	AccountLocation	Read-only
Activity	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
	<a href="#">Audit Fields</a>	Read-only
Address	AddressId	Read-only

Table 218. 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
	<a href="#">Audit Fields</a>	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
		<a href="#">Audit Fields</a>
Campaign	CampaignContactId	Read-only
		<a href="#">Audit Fields</a>
Contact Team	ContactTeamId	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only
	UserRole	Read-only

Table 218. 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 218. 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
	<a href="#">Audit Fields</a>	Read-only

Table 218. 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	
SharedAddress	All fields	Read-only

Table 219 details the status key for the contact object.

Table 219. Status Key for the Contact Object

Child Component	Field Name
Contact (parent)	<a href="#">Audit Fields</a>
	AccountId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Account	<a href="#">Audit Fields</a>
	Name and Location
	AccountExternalId

Table 219. Status Key for the Contact Object

Child Component	Field Name
Activity	<a href="#">Audit Fields</a>
	ActivityExternalId
	ActivityId
	ActivityIntegrationId
Asset	<a href="#">Audit Fields</a>
Attachment	<a href="#">Audit Fields</a>
	Id
	ContactId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId
Campaign	<a href="#">Audit Fields</a>
	CampaignContactId
	CampaignExternalSystemId
	CampaignId
	ModId
ContactNote	<a href="#">Audit Fields</a>
	ContactId
	ExternalSystemId
	Id
	IntegrationId
CustomObject3	CustomObject3Id
	ContactCustomObject3CreatedById
	ContactCustomObject3CreatedDate
	ContactCustomObject3ModifiedById
	ContactCustomObject3ModifiedDate
Interests	<a href="#">Audit Fields</a>
	InterestId
	ExternalSystemId

Table 219. Status Key for the Contact Object

Child Component	Field Name
Lead	<a href="#">Audit Fields</a>
	AccountId
	CampaignId
	ContactId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
Opportunity	<a href="#">Audit Fields</a>
	AccountId
	ExternalSystemId
	IntegrationId
	LeadId
	OpportunityId
RelatedContact	<a href="#">Audit Fields</a>
	ContactRelationshipId
	RelatedContactId
Revenue	<a href="#">Audit Fields</a>
	ExternalId
	IntegrationId
	RevenueId
ServiceRequest	<a href="#">Audit Fields</a>
	AccountId
	ContactID
	ExternalSystemId
	IntegrationId
	ServiceRequestId
SharedAddress	<a href="#">Audit Fields</a>

Table 219. Status Key for the Contact Object

Child Component	Field Name
Team	<a href="#">Audit Fields</a>
	ContactTeamId
	UserExternalSystemId
	UserId
	UserIntegrationId

[Table 220](#) details the pick map fields for the contact object.

Table 220. 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 220. 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 220. 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 220. 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 221 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 221. 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 221. 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 221. 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 221. 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	
	OpportunityId	ExternalSystemId
	Owner	OpportunityId
	Revenue	
	CloseDate	
	Forecast	
	ExpectedRevenue	
	Probability	
	Priority	
	ReasonWonLost	
	SalesStage	
	Status	
	Account	

Table 221. 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 221. 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 221. 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	
Shared Address	None	AddressId
		AddressExternalSystemId

Table 222 details the picklists available for the contact object.

Table 222. 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
Related Contact	Relationship
	Status

Table 222. Picklists Available for the Contact Object

Child Component	Field Name
Revenue	Type
	Status
	Frequency
SharedAddress	AddressType
	CurrencyCode

[Table 223](#) details a number of contact object fields that you must not use for customer integrations.

Table 223. 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 224](#) details the method called by the Current user service, and its name as defined on the service.

Table 224. Methods Called by Current User Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 96</a>	CurrentUserQueryPage



## Fields

[Table 225](#) details the required and read-only fields for the current user object.

Table 225. 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 226](#) 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 226. 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 227](#) details the methods called by the CustomObject1 service. The methods for CustomObject2 and CustomObject3 follow the same pattern.

Table 227. Methods Called by CustomObject1 Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	CustomObject1Delete
<a href="#">"DeleteChild" on page 92</a>	CustomObject1DeleteChild
<a href="#">"Insert" on page 93</a>	CustomObject1Insert
<a href="#">"InsertChild" on page 94</a>	CustomObject1InsertChild

Table 227. Methods Called by CustomObject1 Service

Method	Name as Defined in Service
<a href="#">"InsertOrUpdate" on page 95</a>	CustomObject1OrUpdate
<a href="#">"QueryPage" on page 96</a>	CustomObject1Page
<a href="#">"Update" on page 106</a>	CustomObject1Update
<a href="#">"UpdateChild" on page 107</a>	CustomObject1UpdateChild

## Fields

[Table 228](#) details the required and read-only fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 228. 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
	<a href="#">Audit Fields</a>	Read-only

Table 228. 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
	<a href="#">Audit Fields</a>	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 228. 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 229 details the status key for the CustomObject1 object. The status keys for CustomObject2 and CustomObject3 follow a similar pattern.

Table 229. Status Key for the CustomObject1 Object

Child Component	Field Name
CustomObject1 (parent)	<a href="#">Audit Fields</a>
	CustomObject1Id
	ExternalSystemId
	IntegrationId
Account	CustomObject1AccountId
	CObj1AccountCreatedById
	CObj1AccountCreatedDate
	CObj1AccountModifiedById
	CObj1AccountModifiedDate
Attachment	<a href="#">Audit Fields</a>
	Id
	CustomObjectId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId

Table 229. 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	<a href="#">Audit Fields</a>
	CustomObject1TeamId

Table 230 details the pick map fields for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 230. 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 230. 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 230. Pick Map Fields for the CustomObject1 Object

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

Table 231 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 231. 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 231. Filterable Fields and User Key Fields on the CustomObject1 Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Contact	ContactId	CustomObject1ContactId
	CObj1ContactModifiedById	ExternalSystemId
	CObj1ContactModifiedDate	IntegrationId
	ContactType	
	ExternalSystemId	
	IntegrationId	
CustomObject2	CustomObject2Id	CustomObject2Id
	CObj1CustomObject2ModifiedById	ExternalSystemId
	CObj1CustomObject2ModifiedDate	IntegrationId
	ExternalSystemId	
	IntegrationId	
	Name	
	Type	
CustomObject3	CustomObject3Id	CustomObject3Id
	CObj1CustomObject3ModifiedById	ExternalSystemId
	CObj1CustomObject3ModifiedDate	IntegrationId
	ExternalSystemId	
	IntegrationId	
	Name	
	Type	
Opportunity	CObj1OpportunityModifiedById	OpportunityId
	CObj1OpportunityModifiedDate	ExternalSystemId
	ExternalSystemId	IntegrationId
	IntegrationId	
	OpportunityId	
Portfolio	CObj1PortfolioModifiedById	PortfolioId
	CObj1PortfolioModfiedDate	ExternalSystemId
	ExternalSystemId	IntegrationId
	IntegrationId	
	PortfolioId	

Table 231. 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 232 details the picklists available for the CustomObject1 object. The fields for CustomObject2 and CustomObject3 follow a similar pattern.

Table 232. 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](#), [Opportunity](#), [Team](#)

### 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 233 details the methods called by the Lead service.

Table 233. Methods Called by Lead Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	LeadDelete
<a href="#">"DeleteChild" on page 92</a>	LeadDeleteChild
<a href="#">"Insert" on page 93</a>	LeadInsert
<a href="#">"InsertChild" on page 94</a>	LeadInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	LeadInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	LeadQueryPage
<a href="#">"Update" on page 106</a>	LeadUpdate
<a href="#">"UpdateChild" on page 107</a>	LeadUpdateChild

## Fields

Table 234 details the required and read-only fields for the lead object.

Table 234. 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
	<a href="#">Audit Fields</a>	Read-only
	LastUpdated	Read-only
	LeadConcatField	Read-only
	LeadFullName	Read-only
	ReferredById	Read-only

Table 234. 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
<a href="#">Audit Fields</a>	Read-only	
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	LeadId	Read-only
	Id	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 235 details the status key for the lead object.

Table 235. Status Key for the Lead Object

Child Component	Field Name
Lead (parent)	<a href="#">Audit Fields</a>
	AccountId
	CampaignId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
	OpportunityId
Activity	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	LeadId
Attachment	<a href="#">Audit Fields</a>
	Id
	LeadId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId

Table 236 details the pick map fields for the lead object.

Table 236. 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 236. 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 237 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 237. 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 237. 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	None	PartnerIntegrationId
		PartnerExternalSystemId

Table 238 details the picklists available for the lead object.

Table 238. Picklists Available for the Lead Object

Field Name
AssessmentCategory1 through AssessmentCategory4
Country
IndexedPick0 through IndexedPick5
LeadType
MrMrs
OwnershipStatus
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 239](#) details the methods called by the Note service.

Table 239. Methods Called by Note Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	NoteDelete
<a href="#">"Insert" on page 93</a>	NoteInsert
<a href="#">"InsertOrUpdate" on page 95</a>	NoteInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	NoteQueryPage
<a href="#">"Update" on page 106</a>	NoteUpdate

## Fields

[Table 240](#) details the required and read-only fields for the note object.

Table 240. 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
	<a href="#">Audit Fields</a>	Read-only

[Table 241](#) details the status key for the note object.

Table 241. Status Key for the Note Object

Child Component	Field Name
Note (parent)	<a href="#">Audit Fields</a>
	NotelId

[Table 242](#) provides a list of the filterable fields and a list of user key combinations for the note object.

Table 242. 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 243](#) details the methods called by the Opportunity service.

Table 243. Methods Called by Opportunity Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	OpportunityDelete
<a href="#">"DeleteChild" on page 92</a>	OpportunityDeleteChild
<a href="#">"Insert" on page 93</a>	OpportunityInsert
<a href="#">"InsertChild" on page 94</a>	OpportunityInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	OpportunityInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	OpportunityQueryPage
<a href="#">"Update" on page 106</a>	OpportunityUpdate
<a href="#">"UpdateChild" on page 107</a>	OpportunityUpdateChild

## Fields

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

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

Child Component	Field Name	Type
Opportunity (parent)	AccountId	Required
	CloseDate	Required
	OpportunityName	Required
	SalesStage	Required
	<a href="#">Audit Fields</a>	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
	<a href="#">Audit Fields</a>	Read-only

Table 244. 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
	<a href="#">Audit Fields</a>	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	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 244. 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 245 details the status key for the opportunity object.

Table 245. Status Key for the Opportunity Object

Child Component	Field Name
Opportunity (parent)	<a href="#">Audit Fields</a>
	AccountId
	ExternalSystemId
	Id
	IntegrationId
Activity	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	OpportunityId
Attachment	<a href="#">Audit Fields</a>
	Id
	OpportunityId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId
Competitor	<a href="#">Audit Fields</a>
	OpportunityCompetitorId
	CompetitorId
	CompetitorExternalSystemId
Contact	<a href="#">Audit Fields</a>
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	OpportunityId



Table 245. Status Key for the Opportunity Object

Child Component	Field Name
Lead	<a href="#">Audit Fields</a>
	AccountId
	ContactId
	ExternalSystemId
	IntegrationId
	LastUpdated
	LeadId
	OpportunityId
OpportunityNote	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	OpportunityId
Partner	<a href="#">Audit Fields</a>
	OpportunityPartnerId
	PartnerId
	PartnerExternalSystemId
Product	<a href="#">Audit Fields</a>
	ExternalId
	ProductRevenueId
	IntegrationID

[Table 246](#) details the pick map fields for the opportunity object.

Table 246. 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 246. 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 246. 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 247 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 247. 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 247. 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 247. 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 247. 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 247. 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 248 details the picklists available for the opportunity object.

Table 248. 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 249](#) details the methods called by the Product service.

Table 249. Methods Called by Product Service

Method	Name as Defined in Service
<a href="#">"Insert" on page 93</a>	ProductInsert
<a href="#">"InsertOrUpdate" on page 95</a>	ProductInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	ProductQueryPage
<a href="#">"Update" on page 106</a>	ProductUpdate

## Fields

All fields on the product object are filterable.

[Table 250](#) details the required and read-only fields for the product object.

Table 250. Required and Read-Only Fields for the Product Object

Child Component	Field Name	Type
Product (parent)	ProductName	Required
	<a href="#">Audit Fields</a>	Read-only

[Table 251](#) details the status key for the product object.

Table 251. Status Key for the Product Object

Child Component	Field Name
Product (parent)	<a href="#">Audit Fields</a>
	Id
	IntegrationId

[Table 252](#) details the pick map field for the product object.

Table 252. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 253 details the user keys for the product object.

Table 253. User Keys for the Product Object

Child Component	Field Name
Product (parent)	ProductId
	IntegrationId
	ExternalSystemId

Table 254 details the picklists available for the product object.

Table 254. 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 255](#) details the methods called by the Product category service.

Table 255. Methods Called by Product Category Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	ProductCategoryDelete
<a href="#">"Insert" on page 93</a>	ProductCategoryInsert
<a href="#">"InsertOrUpdate" on page 95</a>	ProductCategoryInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	ProductCategoryQueryPage
<a href="#">"Update" on page 106</a>	ProductCategoryUpdate

## Fields

All fields on the product category object are filterable.

[Table 256](#) details the required and read-only fields for the product category object.

Table 256. Required and Read-Only Fields for the Product Category Object

Child Component	Field Name	Type
ProductCategory	CategoryName	Required
	<a href="#">Audit Fields</a>	Read-only
	ModifiedByFullName	Read-only

Table 257 details the status key for the product category object.

Table 257. Status Key for the Product Category Object

Child Component	Field Name
ProductCategory	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	Name

Table 258 details the pick map field for the product category object.

Table 258. Pick Map Field for the Product Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 259 details the user keys for the product category object.

Table 259. 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](#), [Solution](#), [Team](#)

For information about using attachments with this object, see [Appendix A, "Using Attachments with Web Services On Demand."](#)

### Methods Called

[Table 260](#) details the methods called by the Service request service.

Table 260. Methods Called by Service Request Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	ServiceRequestDelete
<a href="#">"DeleteChild" on page 92</a>	ServiceRequestDeleteChild
<a href="#">"Insert" on page 93</a>	ServiceRequestInsert
<a href="#">"InsertChild" on page 94</a>	ServiceRequestInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	ServiceRequestInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	ServiceRequestQueryPage
<a href="#">"Update" on page 106</a>	ServiceRequestUpdate
<a href="#">"UpdateChild" on page 107</a>	ServiceRequestUpdateChild

## Fields

Table 261 details the required and read-only fields for the service request object.

Table 261. 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
	<a href="#">Audit Fields</a>	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
	<a href="#">Audit Fields</a>	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 262 details the status key for the service request object.

Table 262. Status Key for the Service Request Object

Child Component	Field Name
ServiceRequest (parent)	<a href="#">Audit Fields</a>
	AccountId
	ContactId
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Activity	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
Attachment	<a href="#">Audit Fields</a>
	Id
	SRIId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId
ServiceRequestNote	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	LastUpdated
	ServiceRequestId
Solution	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 263 details the pick map fields for the service request object.

Table 263. 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 264 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 264. 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	
Book	None	BookId
		BookName



Table 264. Filterable Fields and User Key Fields on the Service Request Object's Child Components

Child Component	Filterable Fields	User Key Field Combinations
Service Request Note	Subject	Subject and Description
		ExternalSystemId
Solution	Title	Title
	Published	IntegrationId
	SolutionId	ExternalSystemId
	Status	
	Id	

Table 265 details the picklists available for the service request object.

Table 265. 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 266](#) details the methods called by the Solution service.

Table 266. Methods Called by Solution Service

Method	Name as Defined in Service
<a href="#">“Delete” on page 91</a>	SolutionDelete
<a href="#">“DeleteChild” on page 92</a>	SolutionDeleteChild
<a href="#">“Insert” on page 93</a>	SolutionInsert
<a href="#">“InsertChild” on page 94</a>	SolutionInsertChild
<a href="#">“InsertOrUpdate” on page 95</a>	SolutionInsertOrUpdate
<a href="#">“QueryPage” on page 96</a>	SolutionQueryPage
<a href="#">“Update” on page 106</a>	SolutionUpdate
<a href="#">“UpdateChild” on page 107</a>	SolutionUpdateChild

## Fields

[Table 267](#) details the required and read-only fields for the solution object.

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

Child Component	Field Name	Type
Solution (parent)	Title	Required
	<a href="#">Audit Fields</a>	Read-only
	CreatorId	Read-only
	LastUpdated	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	SolutionId	Read-only
	Id	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 268 details the status key for the solution object.

Table 268. Status Key for the Solution Object

Child Component	Field Name
Solution (parent)	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated
Attachment	<a href="#">Audit Fields</a>
	Id
	SolutionId
ServiceRequest	<a href="#">Audit Fields</a>
	ExternalSystemId
	Id
	IntegrationId
	LastUpdated

Table 269 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 269. 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

Table 269. Filterable Fields and User Key Fields on the Solution Object's Child Components

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

Table 270 details the picklists available for the solution object.

Table 270. 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 271 details the methods called by the Territory service.

Table 271. Methods Called by Territory Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	TerritoryDelete
<a href="#">"Insert" on page 93</a>	TerritoryInsert
<a href="#">"InsertOrUpdate" on page 95</a>	TerritoryInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	TerritoryQueryPage
<a href="#">"Update" on page 106</a>	TerritoryUpdate

## Fields

All fields on the territory object are filterable. The TerritoryName, TerritoryExternalSystemId, and TerritoryIntegrationId fields are user keys for the territory object.

Table 272 details the required and read-only fields for the territory object.

Table 272. Required and Read-Only Fields for the Territory Object

Child Component	Field Name	Type
Territory (parent)	TerritoryName	Required
	Territory	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 273 details the status key for the territory object.

Table 273. Status Key for the Territory Object

Child Component	Field Name
Territory (parent)	<a href="#">Audit Fields</a>
	TerritoryExternalSystemId
	TerritoryIntegrationId
	TerritoryId

Table 274 details the pick map field for the territory object.

Table 274. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

Table 275 details the picklists available for the territory object.

Table 275. 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 276](#) details the methods called by the User service.

Table 276. Methods Called by User Service

Method	Name as Defined in Service
<a href="#">"DeleteChild" on page 92</a>	UserDeleteChild
<a href="#">"Insert" on page 93</a>	UserInsert
<a href="#">"InsertChild" on page 94</a>	UserInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	UserInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	UserQueryPage
<a href="#">"Update" on page 106</a>	UserUpdate
<a href="#">"UpdateChild" on page 107</a>	UserUpdateChild

## Fields

All fields on the user object are filterable.

[Table 277](#) details the required and read-only fields for the user object.

Table 277. 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
	<a href="#">Audit Fields</a>	Read-only
	LastSignInDateTime	Read-only
	ManagerFullName	Read-only

Table 278 details the status key for the user object.

Table 278. Status Key for the User Object

Child Component	Field Name
User	ModifiedById
	ModifiedDate
	EMailAddr
	UserId
	IntegrationId

Table 279 details the pick map field for the user object.

Table 279. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

Table 280 provides a list of user key combinations for the user object.

Table 280. 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 281](#) details the methods called by the User group service.

Table 281. Methods Called by User Group Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	UserGroupDelete
<a href="#">"DeleteChild" on page 92</a>	UserGroupDeleteChild
<a href="#">"Insert" on page 93</a>	UserGroupInsert
<a href="#">"InsertChild" on page 94</a>	UserGroupInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	UserGroupInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	UserGroupQueryPage
<a href="#">"Update" on page 106</a>	UserGroupUpdate
<a href="#">"UpdateChild" on page 107</a>	UserGroupUpdateChild

### Fields

[Table 282](#) details the required and read-only fields for the user group object.

Table 282. Required and Read-Only Fields for the User Group Object

Child Component	Field Name	Type
User Group (parent)	Name	Required
	UserGroupId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 282. Required and Read-Only Fields for the User Group Object

Child Component	Field Name	Type
User	UserGroupUserId	Read-only
	UserId	Read-only
	Alias	Read-only
	Email	Read-only
	Role	Read-only
	UserFirstName	Read-only
	UserLastName	Read-only
	<a href="#">Audit Fields</a>	Read-only

[Table 283](#) details the status key for the user group object.

Table 283. Status Key for the User Group Object

Child Component	Field Name
UserGroup (parent)	<a href="#">Audit Fields</a>
	UserGroupId
	UserGroupIntegrationId
	UserGroupExternalSystemId
User	<a href="#">Audit Fields</a>
	Members_UserId
	UserExternalSystemId
	UserIntegrationId

[Table 284](#) details the pick map field for the user group object.

Table 284. Pick Map Field for the User Group Object

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

Table 285 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 285. 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
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:

- ["Event" on page 492](#)

## Event

The Event object allows you to plan and track medical events such as a medical educational seminar, a symposium, or a customer appreciation event. An event can be as simple as a lunch-and-learn session in a physician's office or as complex as a seminar series or national sales meeting.

**NOTE:** The Event Web service was labelled as the MedEvent Web service before Oracle CRM On Demand Release 25.

## Child Component

[Invitee](#)

## Methods Called

[Table 286](#) details the methods called by the Event service.

Table 286. Methods Called by Event Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	MedEdDelete
<a href="#">"DeleteChild" on page 92</a>	MedEdDeleteChild
<a href="#">"Insert" on page 93</a>	MedEdInsert
<a href="#">"InsertChild" on page 94</a>	MedEdInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	MedEdInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	MedEdQueryPage
<a href="#">"Update" on page 106</a>	MedEdUpdate
<a href="#">"UpdateChild" on page 107</a>	MedEdUpdateChild

## Fields

[Table 287](#) details the read-only fields for the Event object and its child component.

Table 287. Read-Only Fields on the Event Object

Child Component	Field Name	Type
Event (parent)	EndDate	Required
	Name	Required
	Objective	Required
	StartDate	Required
	<a href="#">Audit Fields</a>	Read-only

Table 287. Read-Only Fields on the Event Object

Child Component	Field Name	Type
Invitee	Status	Required
	InviteeId	Read-only
	<a href="#">Audit Fields</a>	Read-only

[Table 288](#) details the status key for the Event object.

Table 288. Status Key for the Event Object

Child Component	Field Name
Event (parent)	<a href="#">Audit Fields</a>
	ExternalId
	MedEdId
Invitee	<a href="#">Audit Fields</a>
	InviteeExternalSystemId
	InviteeId

[Table 289](#) details the pickmap fields for the Event object and its child objects.

Table 289. Pick Map Fields for the Event Object

Child Component	Pick Map Field	Maps To
Event (parent)	ProductExternalId	ProductId
	ProductIntegrationId	ProductId

Table 290 provides a list of the filterable fields and user key combinations for the child components of the Event object.

Table 290. Filterable Fields and User Key Fields on the Event Object's Child Components

Child Components	Filterable Fields	User Key Field Combinations
Event (parent)	ExternalSystemId	MedEdId
	ProductIntegrationId	ExternalSystemId
	ProductId	
	ProductId	
	ProductExternalId	
	PrimaryOwnerId	
Invitee	ContactId	InviteeId
	InviteeExternalSystemId	InviteeExternalSystemId
	Status	
	Type	
	ModifiedDate	

Table 291 details the picklists available for the Event object.

Table 291. Picklists Available for the Event Object

Child Component	Field Name
Event (parent)	EventStatusCode
	EventTypeCode
Invitee	InviteeStatus

For more information on the fields accessible through the Event Web service, go to the Web Services Administration page within the Oracle CRM On Demand application, and generate the WSDL file for the Event 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 495
- "Portfolio" on page 499

## 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](#), [Contact](#), [HouseholdTeam](#)

### Methods Called

[Table 292](#) details the methods called by the Household service.

Table 292. Methods Called by Household Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	HouseholdDelete
<a href="#">"DeleteChild" on page 92</a>	HouseholdDeleteChild
<a href="#">"Insert" on page 93</a>	HouseholdInsert
<a href="#">"InsertChild" on page 94</a>	HouseholdInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	HouseholdInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	HouseholdQueryPage
<a href="#">"Update" on page 106</a>	HouseholdUpdate
<a href="#">"UpdateChild" on page 107</a>	HouseholdInsertChild

## Fields

Table 293 details the required and read-only fields for the household object.

Table 293. 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	
<a href="#">Audit Fields</a>	Read-only	



Table 293. 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
	<a href="#">Audit Fields</a>	Read-only
Contact	ContactId	Required
	ContactExternalId	Required
	ContactIntegrationId	Required
	ContactFirstName	Read-only
	ContactLastName	Read-only
	ContactId	Read-only
	ContactMrMrs	Read-only
	<a href="#">Audit Fields</a>	Read-only
HouseholdTeam	HouseholdAccess	Required
	UserId	Read-only
	UserAlias	Read-only
	UserEmail	Read-only

[Table 294](#) details the status key for the household object.

Table 294. Status Key for the Household Object

Child Component	Field Name
Household (parent)	<a href="#">Audit Fields</a>
	ExternalSystemId
	HouseholdId
	IntegrationID
Book	<a href="#">Audit Fields</a>
	BookId
	ModId
Contact	<a href="#">Audit Fields</a>
	ContactId

Table 294. Status Key for the Household Object

Child Component	Field Name
HouseholdTeam	<a href="#">Audit Fields</a>
	UserExternalSystemId
	HouseholdTeamId
	UserAlias
	UserEmail

Table 295 details the pick map fields for the household object.

Table 295. 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 296 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 296. 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 296. 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 297 details the picklists available for the household object.

Table 297. 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 298](#) details the methods called by the Portfolio service.

Table 298. Methods Called by Portfolio Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	PortfolioDelete
<a href="#">"DeleteChild" on page 92</a>	PortfolioDeleteChild
<a href="#">"Insert" on page 93</a>	PortfolioInsert
<a href="#">"InsertChild" on page 94</a>	PortfolioInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	PortfolioInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	PortfolioQueryPage
<a href="#">"Update" on page 106</a>	PortfolioUpdate
<a href="#">"UpdateChild" on page 107</a>	PortfolioUpdateChild

## Fields

[Table 299](#) details the read-only fields for the portfolio object and its child component.

Table 299. Required and 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
	<a href="#">Audit Fields</a>	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 299. Required and 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
	<a href="#">Audit Fields</a>	Read-only
PortfolioTeam	PortfolioAccess	Required
	UserId	Read-only
	UserAlias	Read-only
	UserEmail	Read-only

[Table 300](#) details the status key for the portfolio object.

Table 300. Status Key for the Portfolio Object

Child Component	Field Name
Portfolio (parent)	<a href="#">Audit Fields</a>
	ExternalSystemId
	PortfolioId
	IntegrationId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId
Contact	<a href="#">Audit Fields</a>
	ContactId
	Id
PortfolioTeam	<a href="#">Audit Fields</a>
	UserId
	UserAlias
	UserEmail
	UserExternalSystemId

Table 301 details the pickmap fields for the portfolio object and its child objects.

Table 301. 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 302 provides a list of the filterable fields and user key combinations for the child components of the portfolio object.

Table 302. 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 302. 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	
	NamedInsured	
	PolicyOwner	
	Relationship	
PortfolioTeam	UserId	UserId
	UserExternalSystemId	UserExternalSystemId
	LastName	UserAlias
	FirstName	UserEmail
	TeamRole	
	PortfolioAccess	

Table 303 details the picklists available for the portfolio object.

Table 303. 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 504
- “Vehicle” on page 507

## 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 304](#) details the methods called by the Dealer service.

Table 304. Methods Called by Dealer Service

Method	Name as Defined in Service
<a href="#">“Delete” on page 91</a>	DealerDelete
<a href="#">“DeleteChild” on page 92</a>	DealerDeleteChild
<a href="#">“Insert” on page 93</a>	DealerInsert
<a href="#">“InsertChild” on page 94</a>	DealerInsertChild
<a href="#">“InsertOrUpdate” on page 95</a>	DealerInsertOrUpdate
<a href="#">“QueryPage” on page 96</a>	DealerQueryPage
<a href="#">“Update” on page 106</a>	DealerUpdate
<a href="#">“UpdateChild” on page 107</a>	DealerUpdateChild



## Fields

All fields on the dealer object are filterable.

Table 305 details the required and read-only fields for the dealer object.

Table 305. 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
	<a href="#">Audit Fields</a>	Read-only
Attachment	DisplayFileName	Required
	FileNameOrURL	Required
	FileDate	Read-only
	FileSize	Read-only
	DealerId	Read-only
	Id	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only
Book	BookName	Required
	BookId	Read-only
	SystemAssociateFlag	Read-only
	ModId	Read-only
	<a href="#">Audit Fields</a>	Read-only

Table 306 details the status key for the dealer object.

Table 306. Status Key for the Dealer Object

Child Component	Field Name
Dealer (parent)	<a href="#">Audit Fields</a>
	DealerId
	DealerIntegrationID
	DealerExternalSystemId
Attachment	<a href="#">Audit Fields</a>
	Id
	DealerId
Book	<a href="#">Audit Fields</a>
	BookId
	ModId

Table 307 details the pick map fields for the dealer object.

Table 307. 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 308 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 308. 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 308. 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 309 details the picklists available for the dealer object.

Table 309. 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 310 details the methods called by the Vehicle service.

Table 310. Methods Called by Vehicle Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 91</a>	VehicleDelete
<a href="#">"DeleteChild" on page 92</a>	VehicleDeleteChild
<a href="#">"Insert" on page 93</a>	VehicleInsert
<a href="#">"InsertChild" on page 94</a>	VehicleInsertChild
<a href="#">"InsertOrUpdate" on page 95</a>	VehicleInsertOrUpdate
<a href="#">"QueryPage" on page 96</a>	VehicleQueryPage
<a href="#">"Update" on page 106</a>	VehicleUpdate
<a href="#">"UpdateChild" on page 107</a>	VehicleUpdateChild

## Fields

Table 311 details the required and read-only fields for the vehicle object.

Table 311. 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
	<a href="#">Audit Fields</a>	Read-only
Contact	ContactId	Required
	ContactExternalSystemId	Required
	ContactIntegrationId	Required
	<a href="#">Audit Fields</a>	Read-only

Table 312 details the status key for the vehicle object.

Table 312. Status Key for the Vehicle Object

Child Component	Field Name
Vehicle (parent)	<a href="#">Audit Fields</a>
	ExternalSystemId
	IntegrationId
	VehicleId
Contact	<a href="#">Audit Fields</a>
	ContactId

Table 313 details the pick map fields for the vehicle object.

Table 313. 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 314 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 314. 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 314. 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 315 details the picklists available for the vehicle object.

Table 315. 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 v1.0. These are objects that are child objects only and are not themselves parent objects:

- "Address" on page 511
- "Attachment" on page 512
- "Audit Trail" on page 512
- "BookUser" on page 512
- "Competitor" on page 512
- "DelegatedUser" on page 513
- "HouseholdTeam" on page 513
- "Interests" on page 514
- "Invitee" on page 514
- "Login History" on page 514
- "Multiple Contact Roles" on page 514
- "OpportunityTeam" on page 514
- "PortfolioTeam" on page 515
- "ProductsDetailed" on page 515
- "PromotionalItemsDropped" on page 515
- "Quota" on page 516
- "Recipient" on page 516
- "Related Account" on page 516
- "Related Contact" on page 516
- "Revenue" on page 516
- "SampleDropped" on page 517
- "Shared Address" on page 517
- "SubBook" on page 517
- "Team" on page 518

## 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.

For information about converting unshared addresses to shared addresses using UI export functionality and Web services calls, see ["Converting Unshared Addresses to Shared Addresses" on page 528](#).

## 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 316](#) details the picklists available for the competitor object.

Table 316. 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 events, including feedback about the invitation.

### Parent Object

[Event](#)

## 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](#), [Opportunity](#)

## 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](#)

## Shared Address

The shared address child object stores information on addresses that can be shared between accounts and contacts that have a relationship with the parent record.

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, and contact records.

For Web Services v1.0, the shared address child object is read only; insert, update, and delete operations are not available. For more information about the shared address functionality, see *Oracle CRM On Demand Online Help*.

When the Enable Shared Addresses check box 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 511](#).

## Parent Objects

[Account](#), [Contact](#)

## 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](#), [Service Request](#), [Lead](#)

# Core Parent Objects (Web Services v2.0)

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:

- [“Partner Relationship Management Edition Objects \(Web Services v2.0\)” on page 581](#)
- [“Life Sciences Edition Parent Objects \(Web Services v2.0\)” on page 613](#)
- [“Financial Services Edition Parent Objects \(Web Services v2.0\)” on page 670](#)
- [“Automotive Edition Parent Objects \(Web Services v2.0\)” on page 697](#)

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

- [“Account” on page 519](#)
- [“Activity” on page 522](#)
- [“Address” on page 527](#)
- [“Address Location” on page 539](#)
- [“Allotment Usage” on page 540](#)
- [“Asset” on page 541](#)
- [“Book” on page 543](#)
- [“Campaign” on page 544](#)
- [“Category” on page 546](#)
- [“Contact” on page 547](#)
- [“CustomObject” on page 551](#)
- [“Division” on page 555](#)
- [“Exchange Rate” on page 557](#)
- [“Group” on page 558](#)

- ["Lead" on page 559](#)
- ["Modification Tracking" on page 562](#)
- ["Note" on page 563](#)
- ["Opportunity" on page 564](#)
- ["Period" on page 566](#)
- ["Product" on page 567](#)
- ["Service Allotment" on page 569](#)
- ["Service Request" on page 571](#)
- ["Social Media" on page 573](#)
- ["Social Profile" on page 574](#)
- ["Solution" on page 576](#)
- ["Territory" on page 577](#)
- ["User" on page 579](#)

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 375](#).

## 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 748](#) and for more information about the address child object, see ["Address" on page 709](#).

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 32.

## Child Objects

[Account Contact](#), [Account Opportunity](#), [Activity](#), [Address](#), [Asset](#), [Attachment](#), [Book](#), [Business Plan](#), [Call Activity History](#), [Claim](#), [Competitor](#), [Contact](#), [Contact Role](#), [Custom Object 1 through 3](#), [Custom Object 4 through 15](#), [Deal Registration](#), [Event](#), [Financial Account](#), [Financial Account Holder](#), [Financial Plan](#), [Image](#), [Lead](#), [Note](#), [Objective](#), [Partner](#), [Plan Account](#), [Policy](#), [Policy Holder](#), [Revenue](#), [Sales Assessment](#), [Sales Assessment Value](#), [Sample Request](#), [Service Request](#), [Shared Address](#), [Special Pricing Product](#), [Team](#)

**NOTE:** When `Enabled Shared Addresses` is selected on the company profile, the address child object (`<ListOfAddress>`) cannot be used for insert, update, or delete operations. Only association and dissociation is allowed for the shared address child object. To update a shared address, you must access the top-level shared address object.



## Methods Called

Table 317 details the methods called by the Account service.

Table 317. Methods Called by Account Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	AccountDelete
<a href="#">"Execute" on page 116</a>	AccountExecute
<a href="#">"Insert" on page 120</a>	AccountInsert
<a href="#">"QueryPage" on page 123</a>	AccountQueryPage
<a href="#">"Update" on page 145</a>	AccountUpdate

## Fields

Table 318 details the required and read-only fields for the account object.

Table 318. Required and Read-Only Fields for the Account Object

Field Name	Type
AccountName	Required
<a href="#">Audit Fields</a>	Read-only

Table 319 details the pick map fields for the account object.

Table 319. Pick Map Fields for the Account Object

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

Table 320 provides a list of the user key combinations for the account object.

Table 320. User Key Fields on the Account Object

User Key Field Combinations
Id

Table 320. User Key Fields on the Account Object

User Key Field Combinations
ExternalSystemId
AccountName and Location

Table 321 details the picklists available for the account object.

Table 321. Picklists Available for the Account Object

Field Name
AccountType
AssessmentFilter1 through AssessmentFilter5
CallFrequency
IndexedPick0 through IndexedPick5
InfluenceType
MarketPotential
MarketSegment
PartnerStatus
Priority
Region
Route
Stage
Status

## 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.

**NOTE:** The CMSId field is accessible in the WSDL file for Activity although it is not available in the Oracle CRM On Demand UI. This field is for internal use only and is not intended for use by customers.

### Creating attendee calls

You can create an attendee call using the Activity service by passing an account call as the parent activity Id.

To create the attendee call, you insert an activity object where the CallType field is set to AttendeeCall and the ParentActivityId field is set to the row Id of an account call.

### 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*.

### Reading Private Activities

Users who have the View Private Activities privilege and the Can Read All Records setting for the Activity record type enabled in their user role can use Web service calls to read all activities that are marked private, regardless of the owner of the activity. The ViewMode argument must have a value of Broadest (which is the default value).

The View Private Activities privilege does not allow a user to update or delete an activity that is marked private and is owned by another user.

### Synchronization of Events and Appointments

If your company administrator enables the Display Events in Calendar setting in the company profile, then Oracle CRM On Demand automatically maintains an associated appointment record for an event record. When a user creates, updates, or deletes an event record through the UI, the Import Assistant, or Web services, Oracle CRM On Demand creates, updates or deletes the associated appointment record. Conversely, Oracle CRM On Demand automatically updates or deletes the original event record whenever you update or delete the corresponding appointment for the event.

The associated appointments are known as shadow appointments, and in Web service calls, are identified with the <EventMirrorApptFlag> element.

For more information about the Display Events in Calendar setting and about synchronization of events and appointments, see *Oracle CRM On Demand Online Help*.

### Child Objects

[Address](#), [Attachment](#), [Attendee Call](#), [Contact](#), [Contact Best Times](#), [Custom Object 1 through 3](#), [Custom Object 4 through 15](#), [Message Response](#), [Products Detailed](#), [Promotional Items Dropped](#), [Sample Dropped](#), [Sample Transaction](#), [Sales Assessment](#), [Sales Assessment Value](#), [Signature](#), [User](#).

## Methods Called

[Table 322](#) details the methods called by the Activity service.

Table 322. Methods Called by Activity Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ActivityDelete
<a href="#">"Execute" on page 116</a>	ActivityExecute
<a href="#">"Insert" on page 120</a>	ActivityInsert
<a href="#">"QueryPage" on page 123</a>	ActivityQueryPage
<a href="#">"Update" on page 145</a>	ActivityUpdate

## Fields

[Table 323](#) details the required and read-only fields for the activity object.

Table 323. Required and Read-Only Fields for the Activity Object

Field Name	Type
Subject	Required
Activity	Required
AccountName	Read-only
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

Table 323. Required and Read-Only Fields for the Activity Object

Field Name	Type
TotalHoldTime	Read-only
ResolutionCode	Read-only
OwnerFullName	Read-only
ServiceRequestNumber	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 324 details the pick map fields for the activity object.

Table 324. Pick Map Fields for the Activity Object

Pick Map Field	Maps To
OwnerAlias	OwnerId
AccountName	AccountId
AccountLocation	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
CampaignExternalSystemId	CampaignId
CampaignIntegrationId	CampaignId
CampaignName	CampaignId
LeadExternalSystemId	LeadId
LeadIntegrationId	LeadId
MedEdEventExternalSystemId	MedEdEventId
MedEdEventIntegrationId	MedEdEventId
OpportunityExternalSystemId	OpportunityId
OpportunityIntegrationId	OpportunityId
PortfolioExternalSystemId	PortfolioId
PortfolioIntegrationId	PortfolioId
ServiceRequestNumber	ServiceRequestId
ServiceRequestExternalSystemId	ServiceRequestId
ServiceRequestIntegrationId	ServiceRequestId
FundRequestExternalSystemId	FundRequestId
FundRequestIntegrationId	FundRequestId

Table 325 provides a list of the user key combinations for the activity object.

Table 325. User Key Fields on the Activity Object

User Key Field Combinations
Description
Id
ExternalSystemId

Table 326 details the picklists available for the activity object.

Table 326. Picklists Available for the Activity Object

Field Name
ActivitySubtype
ApprovalStatus
CallResult
IndexedPick0 through IndexedPick5
Priority
Subtype
RejectReason
ResolutionCode
Status
<b>NOTE:</b> The Status picklist is editable for users whose role includes the Customize Application - Edit Activity 'Status' Picklist privilege.
Type

## Address

The shared address object stores information about addresses that can be shared between account, contact, dealer, and partner records.

### Child Objects

Account, Contact, Custom Object 1 through 3, Custom Object 4 through 15, Dealer, Partner

**NOTE:** For the Account, Contact, Dealer, and Partner child objects, access is read-only.

## 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 748](#).

**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*.

## Converting Unshared Addresses to Shared Addresses

Release 19 and earlier releases of Oracle CRM On Demand used unshared addresses. To convert unshared addresses from earlier releases to shared addresses, you can follow a procedure that uses the export and import functionality of the Oracle CRM On Demand UI as described in *Oracle CRM On Demand Online Help*.

You can also follow a procedure that uses UI export functionality and Web services calls to import data. This involves writing an application to read exported data from a file and then creating the SOAP requests to send to Oracle CRM On Demand. Sample SOAP request are shown in the procedure in this topic. However, it may be more efficient, especially if the conversion is a one-time process, to use the import functionality of the UI rather than Web services calls.

**TIP:** If the data files are large and the Web services or import operations take longer than you expect, then you can use Oracle Data Loader On Demand with the same input files and mapping files that are defined in the conversion procedure.

**Before you begin:** It is assumed that Oracle CRM On Demand Customer Care has upgraded your company to the current release. To complete this procedure, you must be a company administrator who can perform import and export operations, Web services calls, and customize Oracle CRM On Demand. You require the Manage Addresses privilege to enable shared address functionality.

**NOTE:** When moving to shared addresses from unshared addresses, you must use the Web Services v2.0 API for all Web services requests because the Web Services v1.0 API does not support access to shared addresses.



In the following procedure, the steps are performed in the Oracle CRM On Demand UI unless otherwise stated:

*To convert unshared addresses to shared addresses*

- 1 If it is selected, deselect the Enable Shared Addresses setting in the company profile.
- 2 Export the data from the address-related objects that you want to convert to shared addresses.

The list includes the following items:

■ All account data sections including:

- Accounts
- Account Addresses
- Contacts
- Account Partners
- Addresses

■ All contact data sections including:

- Contacts
- Contact Addresses
- Accounts
- Addresses

■ All dealer data sections including:

- Dealer
- Dealer Addresses

■ All partner data sections including:

- Partner
- Partner Addresses

■ All address data

This export operation must export all the relevant account, contact, partner, dealer, and address data from Oracle CRM On Demand to an external file. You must be able to use the external file with the data correction and standardization method that you choose.

- 3 Correct and reformat the data so that importing these addresses into Oracle CRM On Demand does not result in duplicate data and so that the account, contact, dealer, and partner data is set up with the associated addresses.

It is likely that many duplicate and invalid addresses will exist after the export operation. For example, because addresses were not shared in earlier releases, contacts residing at the same location might have identical or similar addresses. However, minor differences might arise because of typing and other errors. For the duplicate addresses that are likely to occur, reimporting the records (for example, accounts) and associating them with newly created shared address records will not delete the preexisting address associations from the earlier release. Consequently, you must do the following:

- Verify your data and then remove duplicate records when preparing the address data that you want to import.
- Take ownership of the preexisting unshared address data associated with your account, contact, dealer, and partner records if you decide to associate those records with the shared address version of the address data. That is, you must remove and disassociate, and choose whether to retain the preexisting unshared address data.

For information on preparing the data for import or use with Web services, see the following topics in *Oracle CRM On Demand Online Help*:

- Address Fields: Import Preparation
- Account Fields: Import Preparation
- Contact Fields: Import Preparation
- Dealer Fields: Import Preparation
- Partner Fields: Import Preparation

- 4 Import the address record data by using Web services calls to insert the Address record data into Oracle CRM On Demand.

Inserting the shared Address top-level record data is a prerequisite for inserting account, contact, dealer, and partner record data. The following is a sample SOAP request for inserting the data:

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns: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-tokenprofile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>
```

```
</wsse: UsernameToken>
</wsse: Security>
</soap: Header>
<soap: Body>
  <SharedAddressInsert_Input xmlns="urn: crmondemand/ws/ecbs/sharedaddress/">
    <ListOfSharedAddress>
      <SharedAddress>
        <AddressName>LS Address 8</AddressName>
        <PrimaryAddressLine1>30 Prospect Ave</PrimaryAddressLine1>
        <Country>USA</Country>
        <City>Hackensack</City>
        <StateProvince>NJ</StateProvince>
        <ZipCode>07601</ZipCode>
      </SharedAddress>
      <SharedAddress>
        <AddressName>LS Address 9</AddressName>
        <PrimaryAddressLine1>100 S Jackson Ave</PrimaryAddressLine1>
        <Country>USA</Country>
        <City>Pittsburgh</City>
        <StateProvince>PA</StateProvince>
        <ZipCode>15202</ZipCode>
      </SharedAddress>
    </ListOfSharedAddress>
  </SharedAddressInsert_Input>
</soap: Body>
</soap: Envelope>
```

Make sure that you retrieve the Address Ids from the SOAP response for the insert operations.

- 5 Import the account record data by using Web services calls to insert the Account-Address record data into Oracle CRM On Demand.

This insertion associates the shared addresses with the accounts. The following is a sample SOAP request for inserting the data. The <AddressId> values used are those retrieved from the response in the previous step:

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns: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:PasswordType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-tokenprofile-1.0#PasswordText">%%PASSWORD%%</wss:Password>

      </wss:UsernameToken>

    </wss:Security>

  </soap:Header>

  <soap:Body>

    <AccountExecute_Input xmlns="urn:crmondemand/ws/ecbs/account/">

      <ListOfAccount>

        <Account operation="skinode">

          <Id>1QA2-11V40X</Id>

          <ListOfSharedAddress>

            <SharedAddress operation="insert">

              <AddressId>1QA2-1TAEXP</AddressId>

              <AddressType>Office</AddressType>

              <PrimaryBillingAddressFlag>true</PrimaryBillingAddressFlag>

              <PrimaryShippingAddressFlag>true</PrimaryShippingAddressFlag>

            </SharedAddress>

          </ListOfSharedAddress>

        </Account>

      </ListOfAccount>

    </AccountExecute_Input>

  </soap:Body>

</soap:Envelope>
```

```

    </Account>
  </ListOfAccount>
</AccountExecute_Input>
</soap: Body>
</soap: Envelope>

```

- 6 Import the contact record data by using Web services calls to insert the Contact-Address record data into Oracle CRM On Demand.

This insertion associates the shared addresses with the contacts. The following is a sample SOAP request for inserting the data:

```

<?xml version="1.0" encoding="utf-16" standalone="no"?>
<soap: Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns: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-tokenprofile-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="skinode">
          <Id>1QA2-1S5XPM</Id>
          <ListOfSharedAddress>
            <SharedAddress operation="insert">
              <AddressId>1QA2-1TAEXP</AddressId>
              <AddressType>Offi ce</AddressType>
            </SharedAddress>
          </ListOfSharedAddress>
        </Contact>
      </ListOfContact>
    </ContactExecute_Input>
  </soap: Body>
</soap: Envelope>

```

```
<PrimaryContactAddressFlag>true</PrimaryContactAddressFlag>
</SharedAddress>
</ListOfSharedAddress>
</Contact>
</ListOfContact>
</ContactExecute_Input>
</soap: Body>
</soap: Envelope>
```

- 7** Import the dealer record data by using Web services calls to insert the Dealer-Address record data into Oracle CRM On Demand.

This insertion associates the shared addresses with the dealers. The following is a sample SOAP request for inserting the data:

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<soap: Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap: Header>
    <wsse: Security>
      <wsse: UsernameToken>
        <wsse: Username>%%USERNAME%%</wsse: Username>
        <wsse: Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-tokenprofile-1.0#PasswordText">%%PASSWORD%%</wsse: Password>
      </wsse: UsernameToken>
    </wsse: Security>
  </soap: Header>
  <soap: Body>
    <DealerExecute_Input xmlns="urn:crmondemand/ws/ecbs/dealer/">
      <ListOfDealer>
        <Dealer operation="skipnode">
          <Id>10A2-1FH4CA</Id>
          <ListOfSharedAddress>
```

```

        <SharedAddress operation="insert">
        <AddressId>10A2-1TAXP</AddressId>
        <AddressType>Office</AddressType>
        <PrimaryBillingAddressFlag>true</PrimaryBillingAddressFlag>
        <PrimaryShippingAddressFlag>true</PrimaryShippingAddressFlag>
        </SharedAddress>
    </ListOfSharedAddress>
</Dealer>
</ListOfDealer>
</DealerExecute_Input>
</soap: Body>
</soap: Envelope>

```

- 8** Import the partner record data by using Web services calls to insert the Partner-Address record data into Oracle CRM On Demand.

This insertion associates the shared addresses with the partners. The following is a sample SOAP request for inserting the data:

```

<?xml version="1.0" encoding="utf-16" standalone="no"?>
<soap: Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns: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-tokenprofile-1.0#PasswordText">%%PASSWORD%%</
wsse: Password>
            </wsse: UsernameToken>
        </wsse: Security>
    </soap: Header>
    <soap: Body>
        <PartnerInsert_Input xmlns="urn:crmondemand/ws/ecbs/partner/">

```

```
<ListOfPartner>
  <Partner>
    <Id>1QA2-1U31RC</Id>
    <ListOfSharedAddress>
      <SharedAddress>
        <AddressId>1QA2-1TAEXP</AddressId>
        <AddressType>Office</AddressType>
        <PrimaryBillingAddressFlag>true</PrimaryBillingAddressFlag>
        <PrimaryShippingAddressFlag>true</PrimaryShippingAddressFlag>
      </SharedAddress>
    </ListOfSharedAddress>
  </Partner>
</ListOfPartner>
</PartnerInsert_Input>
</soap:Body>
</soap:Envelope>
```

- 9 Confirm that the data has been inserted correctly.
- 10 Set up the page layouts for the appropriate roles in your company, and assign the appropriate access to the page layouts for the user's roles.
- 11 Update your users' roles to include access to the Address record type in step 2 of the Role Management wizard.

If you do not complete this step, then users cannot view address records. For example, if they attempt to associate an address record with an account, then an error message is displayed.

- 12 Select the Enable Shared Addresses setting in the company profile to enable shared addresses.
- 13 If your company uses applications that are external to Oracle CRM On Demand to mark addresses as valid, and if you want to use only those validated addresses in address picklists, address searches, and address associations, then select the Validate Shared Addresses setting in the company profile.

**NOTE:** The Validation Date in an address record must be set to mark the address as validated. For more information, see *Oracle CRM On Demand Online Help*.

- 14 (Optional) After you have verified that the inserted data is correct, you can delete the preexisting unshared address data by using Web services calls. The following is a sample SOAP request for deleting the unshared address data:

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



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsse="http://
docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Header>
    <wsse:Security>
      <wsse:UsernameToken>
        <wsse:Username>%%USERNAME%%</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-tokenprofile-1.0#PasswordText">%%PASSWORD%%</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <AccountExecute_Input xmlns="urn:crmondemand/ws/ecbs/account/">
      <ListOfAccount>
        <Account operation="skipnode">
          <Id>10A2-11V40X</Id>
          <ListOfAddress>
            <Address operation="delete">
              <Id>10A2-11V40Z</Id>
            </Address>
            <Address operation="delete">
              <Id>10A2-11V4P1</Id>
            </Address>
          </ListOfAddress>
        </Account>
      </ListOfAccount>
    </AccountExecute_Input>
  </soap:Body>
</soap:Envelope>
```

## Methods Called

Table 327 details the methods called by the Address service.

Table 327. Methods Called by Address Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SharedAddressDelete
<a href="#">"Execute" on page 116</a>	SharedAddressExecute
<a href="#">"Insert" on page 120</a>	SharedAddressInsert
<a href="#">"QueryPage" on page 123</a>	SharedAddressQueryPage
<a href="#">"Update" on page 145</a>	SharedAddressUpdate

## Fields

Table 328 details the required and read-only fields for the shared address object.

Table 328. Required and Read-Only Fields for the Shared Address Object

Field Name	Type
AddressName	Required
GeocodeMatchCode	Read-only
GeocodeMatchVector	Read-only
GeocodeStatus	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 329 details the pick map fields for the shared address object.

Table 329. Pick Maps Fields for the Shared Address Object

Pick Map Field	Maps To
CustomObjectnName	CustomObjectnId
CustomObjectnExternalSystemId	CustomObjectnId

Table 330 details the user keys for the shared address object.

Table 330. User Keys for the Shared Address Object

Field Name
Id
ExternalSystemId
AddressName

Table 331 details the picklists available for the shared address object.

Table 331. Picklists for the Shared Address Object

Field Name
CurrencyCode

## Address Location

The Address Location Web service allows you to return a list of addresses including both shared and unshared addresses.

### Usage

All addresses, both shared and unshared are returned regardless of whether the Enable Shared Addresses check box is selected on the company profile.

### Methods Called

Table 332 details the methods called by the Address Location service.

Table 332. Methods Called by Address Location Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	AddressLocationQueryPage

Table 333 details the pick map fields for the address location object.

Table 333. Pick Maps Fields for the Address Location Object

Pick Map Field	Maps To
CustomObject $n$ Name	CustomObject $n$ Id
CustomObject $n$ ExternalSystemId	CustomObject $n$ Id

Table 334 details the user keys for the address location object.

Table 334. User Keys for the Address Location Object

Field Name
Id
ExternalSystemId
AddressName

## 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.

Various audit fields are also returned. For more information about administration of service allotments, see *Oracle CRM On Demand Online Help*.

## Methods Called

[Table 335](#) details the methods called by the Allotment Usage service.

Table 335. Methods Called by Allotment Usage Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	AllotmentUsageQueryPage

## Fields

[Table 336](#) details the user keys for the allotment usage object.

Table 336. 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 337](#) details the methods called by the Asset service.

Table 337. Methods Called by Asset Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	AssetDelete
<a href="#">"Execute" on page 116</a>	AssetExecute
<a href="#">"Insert" on page 120</a>	AssetInsert

Table 337. Methods Called by Asset Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	AssetQueryPage
<a href="#">"Update" on page 145</a>	AssetUpdate

## Fields

[Table 338](#) details the required and read-only fields for the asset object.

Table 338. 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
<a href="#">Audit Fields</a>	Read-only

[Table 339](#) details the pick map fields for the asset object.

Table 339. 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 340](#) provides a list of the user key combinations for the asset object.

Table 340. User Key Fields on the Asset Object

User Key Field Combinations
Id
ExternalSystemId

Table 340. User Key Fields on the Asset Object

User Key Field Combinations
ProductId
ProductName
ProductExternalId

Table 341 details the picklists available for the asset object.

Table 341. 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 342 details the methods called by the Book service.

Table 342. Methods Called by Book Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	BookDelete
<a href="#">"Execute" on page 116</a>	BookExecute
<a href="#">"Insert" on page 120</a>	BookInsert
<a href="#">"QueryPage" on page 123</a>	BookQueryPage
<a href="#">"Update" on page 145</a>	BookUpdate

## Fields

[Table 343](#) details the required and read-only fields for the book object.

Table 343. Required and Read-Only Fields for the Book Object

Field Name	Type
BookName	Required
BookId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 344](#) details the pick map fields for the book object.

Table 344. Pick Map Fields for the Book Object

Pick Map Field	Maps To
BookName	BookId

[Table 345](#) provides a list of the user key combinations for the book object.

Table 345. User Key Fields on the Book Object

User Key Field Combinations
Id
BookName

[Table 346](#) details the picklists available for the book object.

Table 346. 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, Contact, Custom Object 1 through 3, Financial Account, Financial Plan, Lead, MDF Request, Event, Note, Policy

## Methods Called

Table 347 details the methods called by the Campaign service.

Table 347. Methods Called by Campaign Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CampaignDelete
<a href="#">"Execute" on page 116</a>	CampaignExecute
<a href="#">"Insert" on page 120</a>	CampaignInsert
<a href="#">"QueryPage" on page 123</a>	CampaignQueryPage
<a href="#">"Update" on page 145</a>	CampaignUpdate

## Fields

Table 348 details the required and read-only fields for the campaign object.

Table 348. Required and Read-Only Fields for the Campaign Object

Field Name	Type
CampaignName	Required
SourceCode	Required
<a href="#">Audit Fields</a>	Read-only
CreatedByFullName	Read-only
LastUpdated	Read-only

Table 349 details the pick map field for the campaign object.

Table 349. Pick Map Field for the Campaign Object

Pick Map Field	Maps To
Owner	OwnerId

Table 350 provides a list of the user key combinations for the campaign object.

Table 350. User Key Fields on the Campaign Object

User Key Field Combinations
Id
ExternalSystemId
SourceCode
CampaignName

Table 351 details the picklists available for the campaign object.

Table 351. Picklists Available for the Campaign Object

Field Name
CampaignType
IndexedPick0 through IndexedPick5
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 352 details the methods called by the Category service.

Table 352. Methods Called by Category Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CategoryDelete
<a href="#">"Execute" on page 116</a>	CategoryExecute

Table 352. Methods Called by Category Service

Method	Name as Defined in Service
<a href="#">"Insert" on page 120</a>	CategoryInsert
<a href="#">"QueryPage" on page 123</a>	CategoryQueryPage
<a href="#">"Update" on page 145</a>	CategoryUpdate

## Fields

[Table 353](#) details the required and read-only fields for the category object.

Table 353. Required and Read-Only Fields for the Category Object

Field Name	Type
CategoryName	Required
<a href="#">Audit Fields</a>	Read-only
ModifiedByFullName	Read-only

[Table 354](#) details the pick map field for the category object.

Table 354. Pick Map Field for the Category Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

[Table 355](#) details the user keys for the category object.

Table 355. User Keys for the Category Object

Field Name
Id
ExternalSystemId
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 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 748](#) and for more information about the address child object, see [“Address” on page 709](#).

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 32](#).

## Child Objects

Account, Activity, Address, Attachment, Book, Business Plan, Blocked Product, Call Activity History, Campaign, Claim, Contact Best Times, Contact State License, Coverage, Custom Object 1 through 3, Deal Registration, Financial Account, Financial Account Holder, Financial Plan, HCP Contact Allocation, Household, Image, Interests, Involved Party, Lead, Message Response, Note, Objective, Opportunity, Plan Contact, Policy, Policy Holder, Portfolio, Related Contact, Revenue, Sales Assessment, Sales Assessment Value, Service Request, Signature, Shared Address, Social Profile, Special Pricing Product, Team.

**NOTE:** When Enabled Shared Addresses is selected on the company profile, the address child object (<ListOfAddress>) cannot be used for insert, update, or delete operations. Only association and dissociation is allowed for the shared address child object. To update a shared address, you must access the top-level shared address object.

## Methods Called

Table 356 details the methods called by the Contact service.

Table 356. Methods Called by Contact Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ContactDelete
<a href="#">"Execute" on page 116</a>	ContactExecute
<a href="#">"Insert" on page 120</a>	ContactInsert
<a href="#">"QueryPage" on page 123</a>	ContactQueryPage
<a href="#">"Update" on page 145</a>	ContactUpdate

## Fields

Table 357 details the required and read-only fields for the contact object.

Table 357. Required and Read-Only Fields for the Contact Object

Field Name	Type
ContactFirstName	Required
ContactLastName	Required
AlternateAddressId	Read-only
ContactConcatField	Read-only
ContactFullName	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 357. Required and Read-Only Fields for the Contact Object

Field Name	Type
Manager	Read-only
PrimaryAddressId	Read-only

Table 358 details the pick map fields for the contact object.

Table 358. Pick Map Fields for the Contact Object

Pick Map Field	Maps To
AccountName	AccountId
Owner	AssignedToAlias
SourceCampaignName	SourceCampaignId
ManagerExternalSystemId	ManagerId

Table 359 provides a list of the user key combinations for the contact object.

Table 359. User Key Fields on the Contact Object

User Key Field Combinations
ContactFirstName and ContactLastName
Id
ExternalSystemId

Table 360 details the picklists available for the contact object.

Table 360. Picklists Available for the Contact Object

Field Name
AssessmentFilter1 through AssessmentFilter4
BestTimeToCall
BlockedReasonCode
BuyingRole
CallFrequency
ClientStatus
ContactType

Table 360. Picklists Available for the Contact Object

Field Name
CurrentInvestmentMix
Degree
ExperienceLevel
Gender
IndexedPick0 through IndexedPick5
InvestmentHorizon
LeadSource
LifeEvent
MaritalStatus
MarketPotential
MrMrs
Objective
PrimaryGoal
RiskProfile
Route
Segment
Tier

[Table 361](#) details a number of contact object fields that you must not use for customer integrations.

Table 361. 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.

### Support for Attachments

For Custom Object1 through 40, attachments are accessible through the following fields:

UI Display Name	Integration Tag Name
Attachment	AttachFileFullName
Attachment: Extension	AttachFileExt
Attachment: File Name	AttachFileName
Attachment: Size (in Bytes)	AttachFileSizeDisp

For Custom Object 4 through 40, the <Attachment> tag is used for the actual attachment and is a string type Base64. For Custom Object 1 through 3 however, the <PrimaryAttachment> tag is used for this purpose.

Requests that specify empty values for the following fields remove the associated attachment and clear all attachment-related field values:

- AttachFileName
- AttachFileExt
- Attachment

For Custom Object 1 through 3 you can also exchange attachments through the Attachment child object as described in [Appendix A, "Using Attachments with Web Services On Demand."](#) For Custom Object 4 through 40, an Attachment child object is not accessible.

### Child Objects

[Account](#), [Accreditation](#), [Activity](#), [Asset](#), [Attachment](#), [Book](#), [Certification](#), [Contact](#), [Course](#), [CustomObject](#), [Custom Object 1 Child](#), [Deal Registration](#), [Exam](#), [Financial Account](#), [Inventory Period](#), [MDF Request](#), [Note](#), [Opportunity](#), [Policy](#), [Portfolio](#), [Sample Lot](#), [Sample Transaction](#), [Special Pricing Request](#), [Team](#).



## Methods Called

Table 362 details the methods called by the CustomObject1 service. The methods for the other Custom Objects follow the same pattern.

Table 362. Methods Called by CustomObject1 Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CustomObject1Delete
<a href="#">"Execute" on page 116</a>	CustomObject1Execute
<a href="#">"Insert" on page 120</a>	CustomObject1Insert
<a href="#">"QueryPage" on page 123</a>	CustomObject1QueryPage
<a href="#">"Update" on page 145</a>	CustomObject1Update

## Fields

Table 363 details the required and read-only fields for the CustomObject1 object.

**NOTE:** This section details the fields for the CustomObject1 objects. The other Custom Objects follow the same pattern.

Table 363. Required and Read-Only Fields for the CustomObject1 Object

Field Name	Type
Id	Required
CustomObject1ExternalSystemID	Required
CustomObject1IntegrationId	Required
OwnerFullName	Read-only.

Table 364 details the pick map fields for the CustomObject1 object.

Table 364. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
AccountExternalId	AccountId
AccountIntegrationId	AccountId
AccountName	AccountId
ActivityExternalId	ActivityId
ActivityIntegrationId	ActivityId
ActivityName	ActivityId

Table 364. Pick Map Fields for the CustomObject1 Object

Pick Map Field	Maps To
CampaignExternalId	CampaignId
CampaignIntegrationId	CampaignId
CampaignName	CampaignId
ContactExternalId	ContactId
ContactFirstName	ContactId
ContactFullName	ContactId
ContactIntegrationId	ContactId
ContactLastName	ContactId
CustomObject $n$ ExternalId	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
OpportunityName	OpportunityId
ParentExternalSystemId	ParentId
ParentIntegrationId	ParentId
PortfolioAccountNumber	PortfolioId
ProductExternalId	ProductId
ProductIntegrationId	ProductId
ProductName	ProductId
SolutionExternalId	SolutionId

Table 364. Pick Map Fields for the CustomObject1 Object

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

Table 365 provides a list of the user key combinations for the CustomObject1 object.

Table 365. User Key Fields on the CustomObject1 Object

User Key Field Combinations
Id
ExternalSystemId

## Division

The division object organizes users into groupings to meet business needs. For example, you can use divisions to organize users on a geographical basis.

### Usage

You can associate divisions with picklist value groups, so that for users in certain divisions, the values that appear in the picklists in certain fields on records are limited to subsets of the values that are defined for the fields. Each division can be associated with one picklist value group only. The PickValueGroupFullName and PickValueGroupID are used to identify the picklist value group.

A user can be associated with one or more divisions. You can associate users with divisions in two ways:

- From a division record, through the user child object
- From a user record, through the division child object

For more information about divisions and picklist value groups, see *Oracle CRM On Demand Online Help*.

### Child Object

User

## Methods Called

Table 366 details the methods called by the Division service.

Table 366. Methods Called by Division Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	DivisionDelete
<a href="#">"Execute" on page 116</a>	DivisionExecute
<a href="#">"Insert" on page 120</a>	DivisionInsert
<a href="#">"QueryPage" on page 123</a>	DivisionQueryPage
<a href="#">"Update" on page 145</a>	DivisionUpdate

## Fields

Table 367 details the required and read-only fields for the division object.

Table 367. Required and Read-Only Fields for the Division Object

Field Name	Type
Name	Required
Id	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 368 details the pick map fields for the division object.

Table 368. Pick Map Fields for the Division Object

Pick Map Field	Maps To
Name	Id

Table 369 provides a list of the user key combinations for the division object.

Table 369. User Key Fields on the Division Object

User Key Field Combinations
Id
Name

## 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 370 details the methods called by the Exchange Rate service.

Table 370. Methods Called by Exchange Rate Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ExchangeRateDelete
<a href="#">"Execute" on page 116</a>	ExchangeRateExecute
<a href="#">"Insert" on page 120</a>	ExchangeRateInsert
<a href="#">"QueryPage" on page 123</a>	ExchangeRateQueryPage
<a href="#">"Update" on page 145</a>	ExchangeRateUpdate

### Fields

Table 371 details the required and read-only fields for the exchange rate object.

Table 371. Required and Read-Only Fields for the Exchange Rate Object

Field Name	Type
FromCurrencyCode	Required
ToCurrencyCode	Required

Table 371. Required and Read-Only Fields for the Exchange Rate Object

Field Name	Type
ExchangeDate	Required
ExchangeRate	Required

Table 372 provides a list of the user key combinations for the exchange rate object.

Table 372. User Key Fields on the Exchange Rate Object

User Key Field Combinations
FromCurrencyCode
ToCurrencyCode
ExchangeDate

## Group

The group object allows you to create groups to which users can be added. Users can only be a member of one group, and groups can contain many users.

The group object is equivalent to the Web Services V1.0 user group object.

## Methods Called

Table 373 details the methods called by the Group service.

Table 373. Methods Called by Group Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	GroupDelete
<a href="#">"Execute" on page 116</a>	GroupExecute
<a href="#">"Insert" on page 120</a>	GroupInsert
<a href="#">"QueryPage" on page 123</a>	GroupQueryPage
<a href="#">"Update" on page 145</a>	GroupUpdate

## Fields

[Table 374](#) details the required and read-only fields for the group object.

Table 374. Required and Read-Only Fields for the Group Object

Field Name	Type
Name	Required
UserGroupId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 375](#) details the pick map field for the group object.

Table 375. Pick Map Field for the Group Object

Pick Map Field	Maps To
UserIntegrationId	UserId
UserExternalSystemId	UserId

[Table 376](#) provides a list of the user key combinations for the group object.

Table 376. 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 32.

## Child Objects

[Activity](#), [Attachment](#), [Book](#), [Custom Object 1 through 3](#), [Custom Object 4 through 15](#), [Deal Registration](#), [Event](#), [Team](#).

## Methods Called

Table 377 details the methods called by the Lead service.

Table 377. Methods Called by Lead Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	LeadDelete
<a href="#">"Execute" on page 116</a>	LeadExecute
<a href="#">"Insert" on page 120</a>	LeadInsert
<a href="#">"QueryPage" on page 123</a>	LeadQueryPage
<a href="#">"Update" on page 145</a>	LeadUpdate

## Fields

Table 378 details the required and read-only fields for the lead object.

Table 378. Required and Read-Only Fields for the Lead Object

Field Name	Type
FirstName	Required
LastName	Required
LeadOwner	Required
ContactFullName	Read-only
<a href="#">Audit Fields</a>	Read-only
LastUpdated	Read-only
LeadConcatField	Read-only
LeadFullName	Read-only
ReferredById	Read-only

Table 379 details the pick map fields for the lead object.

Table 379. Pick Map Fields for the Lead Object

Pick Map Field	Maps To
Campaign	CampaignId
OpportunityName	OpportunityId
Owner	OwnerId
AccountExternalSystemId	AccountId



Table 379. Pick Map Fields for the Lead Object

Pick Map Field	Maps To
OpportunityExternalSystemId	OpportunityId
ContactExternalSystemId	ContactId
CampaignExternalSystemId	CampaignId
ReferredByExternalSystemId	ReferredById

Table 380 provides a list of the user key combinations of the lead object.

Table 380. User Key Fields on the Lead Object

User Key Field Combinations
Id
ExternalSystemId
LeadFirstName and LeadLastName

Table 381 details the picklists available for the lead object.

Table 381. Picklists Available for the Lead Object

Field Name
AssessmentCategory1 through AssessmentCategory4
IndexedPick0 through IndexedPick5
LeadType
MrMrs
OwnershipStatus
Rating
RejectCode
Source

## Modification Tracking

Modification tracking is available for applications developed by Oracle CRM On Demand partners and customer integrations. This includes the Oracle CRM On Demand Disconnected Mobile Sales application. Administrators whose role includes the Manage Modification Tracking privilege can do the following in the UI.

- Enable automatic modification tracking for a given set of record types. For the record types that are tracked, change events are written to a modification tracking record.
- Create event records manually.

Old modification tracking records (records that are 30 days or older) are permanently deleted whenever a regular purge of deleted items occurs in Oracle CRM On Demand.

You can use the Modification Tracking service to query event records or to insert and delete records, if your role includes the Manage Modification Tracking privilege. Requests must include the correct record type name in the `ObjectName` field. For information about the record types supported for modification tracking, and the correct record type names to use in Web service requests, refer to *Oracle CRM On Demand Online Help* and supplemental documentation.

## Child Objects

None.

## Methods Called

[Table 382](#) details the methods called by the Modification Tracking service.

Table 382. Methods Called by Modification Tracking Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ModificationTrackingDelete
<a href="#">"Execute" on page 116</a>	ModificationTrackingExecute
<a href="#">"Insert" on page 120</a>	ModificationTrackingInsert
<a href="#">"QueryPage" on page 123</a>	ModificationTrackingQueryPage
<a href="#">"Update" on page 145</a>	ModificationTrackingUpdate

## Fields

[Table 383](#) details the required and read-only fields for the modification tracking object.

Table 383. Required and Read-Only Fields for the Modification Tracking Object

Field Name	Type
ObjectId	Required
EventName	Required
ObjectName	Required
ModificationNumber	Required
<a href="#">Audit Fields</a>	Read-only

[Table 384](#) provides a list of the user key combinations for the modification tracking object.

Table 384. User Key Fields on the Modification Tracking Object

User Key Field Combinations
Id
ObjectName

## 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 385](#) details the methods called by the Note service.

Table 385. Methods Called by Note Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	NoteDelete
<a href="#">"Execute" on page 116</a>	NoteExecute
<a href="#">"Insert" on page 120</a>	NoteInsert

Table 385. Methods Called by Note Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	NoteQueryPage
<a href="#">"Update" on page 145</a>	NoteUpdate

## Fields

[Table 386](#) details the required and read-only fields for the note object.

Table 386. Required and Read-Only Fields for the Note Object

Field Name	Type
Subject	Required
NotelD	Read-only
OwnerID	Read-only
OwnerAlias	Read-only
ParentNotelD	Read-only
SourceID	Read-only
SourceName	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 387](#) provides a list of the user key combinations for the note object.

Table 387. User Key Fields on the Note Object

User Key Field Combinations
NotelD

## 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](#), [Book](#), [Competitor](#), [Contact](#), [Contact Role](#), [Custom Object 1 through 3](#), [Custom Object 4 through 15](#), [Deal Registration](#), [Event](#), [Lead](#), [Note](#), [Opportunity Child](#), [Products Detailed](#), [Partner](#), [Plan Opportunity](#), [Revenue](#), [Sales Assessment](#), [Sales Assessment Value](#), [Service Request](#), [Special Pricing Product](#), [Team](#).

## Methods Called

[Table 388](#) details the methods called by the Opportunity service.

Table 388. Methods Called by Opportunity Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	OpportunityDelete
<a href="#">"Execute" on page 116</a>	OpportunityExecute
<a href="#">"Insert" on page 120</a>	OpportunityInsert
<a href="#">"QueryPage" on page 123</a>	OpportunityQueryPage
<a href="#">"Update" on page 145</a>	OpportunityUpdate

## Fields

[Table 389](#) details the required and read-only fields for the opportunity object.

Table 389. Required and Read-Only Fields for the Opportunity Object

Field Name	Type
AccountId	Required
CloseDate	Required
OpportunityName	Required
SalesStage	Required
<a href="#">Audit Fields</a>	Read-only
LastUpdated	Read-only
OpportunityConcatField	Read-only

[Table 390](#) details the pick map fields for the opportunity object.

Table 390. Pick Map Fields for the Opportunity Object

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

Table 391 provides a list of the user key combinations for the opportunity object.

Table 391. User Key Fields on the Opportunity Object

User Key Field Combinations
Id
ExternalSystemId
Name

Table 392 details the picklists available for the opportunity object.

Table 392. Picklists Available for the Opportunity Object

Field Name
ApprovalStatus
AssessmentFilter1 through AssessmentFilter4
IndexedPick0 through IndexedPick5
LeadSource
OpportunityType
OwnershipStatus
Priority
Probability
ReasonWonLost
RegistrationStatus
Status
Type
Year

## Period

The period object stores information about the fiscal periods for a company.

## Child Objects

[Business Plan](#), [Messaging Plan](#), [Objective](#)

## Usage

There are various elements of the ListOfFiscalPeriod argument that specify the period data:

- **PeriodName.** The name of the fiscal period.
- **StartDate.** The start date for the fiscal period.
- **EndDate.** The end date for the fiscal period.
- **PeriodType.** The period type.
- **Description.** A description for the fiscal period.

For more information about administration of fiscal periods, see *Oracle CRM On Demand Online Help*.

## Methods Called

[Table 393](#) details the methods called by the Period service.

Table 393. Methods Called by Period Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	FiscalPeriodQueryPage

## Fields

[Table 394](#) details the user keys for the Period object.

Table 394. User Keys for the Period Object

Field Name
PeriodName

## 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

[Image](#), [Price List Line Item](#)

## Methods Called

Table 395 details the methods called by the Product service.

Table 395. Methods Called by Product Service

Method	Name as Defined in Service
<a href="#">"Execute" on page 116</a>	ProductExecute
<a href="#">"Insert" on page 120</a>	ProductInsert
<a href="#">"QueryPage" on page 123</a>	ProductQueryPage
<a href="#">"Update" on page 145</a>	ProductUpdate

## Fields

Table 396 details the required and read-only fields for the product object.

Table 396. Required and Read-Only Fields for the Product Object

Field Name	Type
ProductName	Required
<a href="#">Audit Fields</a>	Read-only

Table 397 details the pick map field for the product object.

Table 397. Pick Map Field for the Product Object

Pick Map Field	Maps To
ParentCategory	ParentCategoryId

Table 398 details the user keys for the product object.

Table 398. User Keys for the Product Object

Field Name
ProductName
Id
ExternalSystemId



Table 399 details the picklists available for the product object.;

Table 399. 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, as follows:

- **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, which consists of the following:
  - **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 400](#) details the methods called by the Service Allotment service.

Table 400. Methods Called by Service Allotment Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	ServiceAllotmentQueryPage

## Fields

[Table 401](#) details the user keys for the service allotment object.

Table 401. 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

[Activity](#), [Attachment](#), [Custom Object 1 through 3](#), [Note](#), [Sales Assessment](#), [Sales Assessment Value](#), [Team](#).

## Methods Called

[Table 402](#) details the methods called by the Service request service.

Table 402. Methods Called by Service Request Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ServiceRequestDelete
<a href="#">"Execute" on page 116</a>	ServiceRequestExecute
<a href="#">"Insert" on page 120</a>	ServiceRequestInsert
<a href="#">"QueryPage" on page 123</a>	ServiceRequestQueryPage
<a href="#">"Update" on page 145</a>	ServiceRequestUpdate

## Fields

[Table 403](#) details the required and read-only fields for the service request object.

Table 403. Required and Read-Only Fields for the Service Request Object

Field Name	Type
ContactEmail	Read-only
ContactFirstName	Read-only

Table 403. Required and Read-Only Fields for the Service Request Object

Field Name	Type
ContactFullName	Read-only
ContactLastName	Read-only
<a href="#">Audit Fields</a>	Read-only
LastUpdated	Read-only
ServiceRequestContactId	Read-only

[Table 404](#) details the pick map fields for the service request object.

Table 404. Pick Map Fields for the Service Request Object

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

[Table 405](#) provides a list of the user key combinations for the service request object.

Table 405. User Key Fields on the Service Request Object

User Key Field Combinations
Id
ExternalSystemId
SRNumber

[Table 406](#) details the picklists available for the service request object.

Table 406. Picklists Available for the Service Request Object

Field Name
Area
Cause
Priority
Source

Table 406. Picklists Available for the Service Request Object

Field Name
Status
Type

## Social Media

The social media object stores information about the social media postings of contacts.

**NOTE:** The social media object is accessible only to users whose role has the Integration:SEM privileges enabled.

## Child Objects

[Service Request](#)

## Methods Called

[Table 407](#) details the methods called by the Social Media service.

Table 407. Methods Called by Social Media Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SocialMediaDelete
<a href="#">"Execute" on page 116</a>	SocialMediaExecute
<a href="#">"Insert" on page 120</a>	SocialMediaInsert
<a href="#">"QueryPage" on page 123</a>	SocialMediaQueryPage
<a href="#">"Update" on page 145</a>	SocialMediaUpdate

## Fields

[Table 408](#) details the required and read-only fields for the social media object.

Table 408. Required and Read-Only Fields for the Social Media Object

Field Name	Type
AuthorId	Required
Community	Required
PostLink	Required

Table 408. Required and Read-Only Fields for the Social Media Object

Field Name	Type
UniqueID	Required
<a href="#">Audit Fields</a>	Read-only

[Table 409](#) details the pick map fields for the social media object.

Table 409. Pick Map Fields for the Social Media Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactIntegrationId	ContactId
ContactExternalSystemId	ContactId

[Table 410](#) provides a list of the user key combinations for the social media object.

Table 410. User Key Fields on the Social Media Object

User Key Field Combinations
Id
ExternalSystemId

## Social Profile

The social profile object stores information about the profiles that contacts use in their social media interactions.

**NOTE:** The social profile object is accessible only to users whose role has the Integration:SEM privileges enabled.

## Child Objects

None.

## Methods Called

[Table 411](#) details the methods called by the Social Profile service.

Table 411. Methods Called by Social Profile Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SocialProfileDelete
<a href="#">"Execute" on page 116</a>	SocialProfileExecute
<a href="#">"Insert" on page 120</a>	SocialProfileInsert
<a href="#">"QueryPage" on page 123</a>	SocialProfileQueryPage
<a href="#">"Update" on page 145</a>	SocialProfileUpdate

## Fields

[Table 412](#) details the required and read-only fields for the social profile object.

Table 412. Required and Read-Only Fields for the Social Profile Object

Field Name	Type
ContactId	Required
AuthorId	Required
Community	Required
Active	Required
<a href="#">Audit Fields</a>	Read-only

[Table 413](#) details the pick map fields for the social profile object.

Table 413. Pick Map Fields for the Social Profile Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactIntegrationId	ContactId
ContactExternalSystemId	ContactId

Table 414 provides a list of the user key combinations for the social profile object.

Table 414. User Key Fields on the Social Profile Object

User Key Field Combinations
Id
ExternalSystemId

## 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 Objects

[Attachment](#), [Book](#), [Message Response](#), [Messaging Plan](#), [Messaging Plan Item](#), [Messaging Plan Item Relation](#), [Service Request](#).

## Methods Called

Table 415 details the methods called by the Solution service.

Table 415. Methods Called by Solution Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SolutionDelete
<a href="#">"Execute" on page 116</a>	SolutionExecute
<a href="#">"Insert" on page 120</a>	SolutionInsert
<a href="#">"QueryPage" on page 123</a>	SolutionQueryPage
<a href="#">"Update" on page 145</a>	SolutionUpdate

## Fields

Table 416 details the required and read-only fields for the solution object.

Table 416. Required and Read-Only Fields for the Solution Object

Field Name	Type
Title	Required
<a href="#">Audit Fields</a>	Read-only



Table 416. Required and Read-Only Fields for the Solution Object

Field Name	Type
CreatorId	Read-only
LastUpdated	Read-only

Table 417 provides a list of the user key combinations for the solution object.

Table 417. User Key Fields on the Solution Object

User Key Field Combinations
Name
Id
ExternalSystemId

Table 418 details the picklists available for the solution object.

Table 418. 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 419 details the methods called by the Territory service.

Table 419. Methods Called by Territory Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	TerritoryDelete
<a href="#">"Execute" on page 116</a>	TerritoryExecute
<a href="#">"Insert" on page 120</a>	TerritoryInsert
<a href="#">"QueryPage" on page 123</a>	TerritoryQueryPage
<a href="#">"Update" on page 145</a>	TerritoryUpdate

## Fields

Table 420 details the required and read-only fields for the territory object.

Table 420. Required and Read-Only Fields for the Territory Object

Field Name	Type
TerritoryName	Required
Territory	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 421 details the pick map field for the territory object.

Table 421. Pick Map Field for the Territory Object

Pick Map Field	Maps To
ParentTerritoryIntegrationId	ParentTerritoryId
ParentTerritoryExternalSystemId	ParentTerritoryId
ParentTerritoryId	ParentTerritoryId

Table 422 details the user key for the territory object.

Table 422. User Key for the Territory Object

Field Name
Id

Table 422. User Key for the Territory Object

Field Name
ExternalSystemId
TerritoryName

[Table 423](#) details the picklist fields available for the territory object.

Table 423. 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.

Starting with Oracle CRM On Demand Release 23, you can use the DefaultBookId and DefaultAnalyticsBookId fields respectively to access the default book and default analytics book for a user. In previous releases it was possible to access the default books through the DefaultBookName and DefaultAnalyticsBookName fields only.

A user can be associated with one or more divisions. You can associate users with divisions in two ways:

- From a division record, through the user child object
- From a user record, through the division child object

The first division that is associated with a user record is automatically specified as the primary division for the user, but you can change the primary division for the user through the PrimaryDivisionName field.

### Child Objects

[Delegated User](#), [Division](#), [User Default Book Per Type](#)

## Methods Called

Table 424 details the methods called by the User service.

Table 424. Methods Called by User Service

Method	Name as Defined in Service
<a href="#">"Execute" on page 116</a>	UserExecute
<a href="#">"Insert" on page 120</a>	UserInsert
<a href="#">"QueryPage" on page 123</a>	UserQueryPage
<a href="#">"Update" on page 145</a>	UserUpdate

## Fields

Table 425 details the required and read-only fields for the user object.

Table 425. Required and Read-Only Fields for the User Object

Field Name	Type
FirstName	Required
LastName	Required
UserLoginId	Required
UserSignInId	Required
Alias	Required
EmailAddr	Required
Role	Required
Status	Required
<a href="#">Audit Fields</a>	Read-only
LastSignInDateTime	Read-only
ManagerFullName	Read-only

Table 426 details the pick map field for the user object.

Table 426. Pick Map Field for the User Object

Pick Map Field	Maps To
Role	RoleId

Table 427 provides a list of user key combinations for the user object.

Table 427. User Key Fields on the User Object

User Key Field Combinations
Id
UserId
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 582
- "Accreditation Request" on page 584
- "Application" on page 585
- "Certification" on page 587
- "Certification Request" on page 589
- "Course" on page 591
- "Course Enrollment" on page 593
- "Deal Registration" on page 595
- "Exam" on page 597
- "Exam Registration" on page 599
- "Fund" on page 601
- "MDF Request" on page 603
- "Partner" on page 605
- "Price List" on page 607
- "Price List Line Item" on page 608
- "Special Pricing Product" on page 610

■ [“Special Pricing Request” on page 611](#)

## 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](#), [Team](#).

### Methods Called

[Table 428](#) details the methods called by the Accreditation service.

Table 428. Methods Called by Accreditation Service

Method	Name as Defined in Service
<a href="#">“Delete” on page 112</a>	AccreditationDelete
<a href="#">“Execute” on page 116</a>	AccreditationExecute
<a href="#">“Insert” on page 120</a>	AccreditationInsert
<a href="#">“QueryPage” on page 123</a>	AccreditationQueryPage
<a href="#">“Update” on page 145</a>	AccreditationUpdate

### Fields

[Table 429](#) details the required and read-only fields for the accreditation object.

Table 429. Required and Read-Only Fields for the Accreditation Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only

Table 430 details the pick map fields for the accreditation object.

Table 430. Pick Map Fields for the Accreditation Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
PrimaryProductCategoryName	PrimaryProductCategoryId
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId

Table 431 provides a list of the user key combinations for the accreditation object.

Table 431. User Key Fields on the Accreditation Object

User Key Field Combinations
Id
Name

Table 432 details the picklists available for the accreditation object.

Table 432. 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 433](#) details the methods called by the Accreditation Request service.

Table 433. Methods Called by Accreditation Request Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	AccreditationRequestDelete
<a href="#">"Execute" on page 116</a>	AccreditationRequestExecute
<a href="#">"Insert" on page 120</a>	AccreditationRequestInsert
<a href="#">"QueryPage" on page 123</a>	AccreditationRequestQueryPage
<a href="#">"Update" on page 145</a>	AccreditationRequestUpdate

### Fields

[Table 434](#) details the required and read-only fields for the accreditation request object.

Table 434. Required and Read-Only Fields for the Accreditation Request Object

Field Name	Type
AccreditationId	Required
PartnerId	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only



Table 435 details the pick map fields for the accreditation request object.

Table 435. 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 436 provides a list of the user key combinations for the accreditation request object.

Table 436. User Key Fields on the Accreditation Request Object

User Key Field Combinations
Id
AccreditationId, PartnerId and RequestDate

Table 437 details the picklists available for the accreditation request object.

Table 437. 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](#), [Team](#).

## Methods Called

Table 438 details the methods called by the Application service.

Table 438. Methods Called by Application Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ApplicationDelete
<a href="#">"Execute" on page 116</a>	ApplicationExecute
<a href="#">"Insert" on page 120</a>	ApplicationInsert
<a href="#">"QueryPage" on page 123</a>	ApplicationQueryPage
<a href="#">"Update" on page 145</a>	ApplicationUpdate

## Fields

Table 439 details the required and read-only fields for the application object.

Table 439. Required and Read-Only Fields for the Application Object

Field Name	Type
ContactFirstName	Required
ContactLastName	Required
SubmissionStatus	Required
CompanyName	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 440 details the pick map fields for the application object.

Table 440. 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 441 provides a list of the user key combinations for the application object.

Table 441. User Key Fields on the Application Object

User Key Field Combinations
Id
ExternalSystemId
CompanyName

Table 442 details the picklists available for the application object.

Table 442. 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](#), [Team](#).

## Methods Called

[Table 443](#) details the methods called by the Certification service.

Table 443. Methods Called by Certification Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CertificationDelete
<a href="#">"Execute" on page 116</a>	CertificationExecute
<a href="#">"Insert" on page 120</a>	CertificationInsert
<a href="#">"QueryPage" on page 123</a>	CertificationQueryPage
<a href="#">"Update" on page 145</a>	CertificationUpdate

## Fields

[Table 444](#) details the required and read-only fields for the certification object.

Table 444. Required and Read-Only Fields for the Certification Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 445](#) details the pick map fields for the certification object.

Table 445. Pick Map Fields for the Certification Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
PrimaryProductCategoryName	PrimaryProductCategoryId

Table 445. Pick Map Fields for the Certification Object

Pick Map Field	Maps To
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId

Table 446 provides a list of the user key combinations for the certification object.

Table 446. User Key Fields on the Certification Object

User Key Field Combinations
Id
Name

Table 447 details the picklists available for the certification object.

Table 447. 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 448](#) details the methods called by the Certification Request service.

Table 448. Methods Called by Certification Request Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CertificationRequestDelete
<a href="#">"Execute" on page 116</a>	CertificationRequestExecute
<a href="#">"Insert" on page 120</a>	CertificationRequestInsert
<a href="#">"QueryPage" on page 123</a>	CertificationRequestQueryPage
<a href="#">"Update" on page 145</a>	CertificationRequestUpdate

## Fields

[Table 449](#) details the required and read-only fields for the certification request object.

Table 449. Required and Read-Only Fields for the Certification Request Object

Field Name	Type
CertificationId	Required
ContactId	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 450](#) details the pick map fields for the certification request object.

Table 450. 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 451 provides a list of the user key combinations for the certification request object.

Table 451. User Key Fields on the Certification Request Object

User Key Field Combinations
Id
CertificationId, ContactId, and RequestDate

Table 452 details the picklists available for the certification request object.

Table 452. 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](#), [Team](#).

## Methods Called

Table 453 details the methods called by the Course service.

Table 453. Methods Called by Course Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CourseDelete
<a href="#">"Execute" on page 116</a>	CourseExecute
<a href="#">"Insert" on page 120</a>	CourseInsert

Table 453. Methods Called by Course Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	CourseQueryPage
<a href="#">"Update" on page 145</a>	CourseUpdate

## Fields

[Table 454](#) details the required and read-only fields for the course object.

Table 454. Required and Read-Only Fields for the Course Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 455](#) details the pick map fields for the course object.

Table 455. 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



Table 455. Pick Map Fields for the Course Object

Pick Map Field	Maps To
ExamType, ExamStatus, ExamName, ExamID, ExamExternalSystemId	ExamId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 456 provides a list of the user key combinations for the course object.

Table 456. User Key Fields on the Course Object

User Key Field Combinations
Id
CourseId, CandidateId, and EnrollDate

Table 457 details the picklists available for the course object.

Table 457. 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 458](#) details the methods called by the Course Enrollment service.

Table 458. Methods Called by Course Enrollment Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CourseEnrollmentDelete
<a href="#">"Execute" on page 116</a>	CourseEnrollmentExecute
<a href="#">"Insert" on page 120</a>	CourseEnrollmentInsert
<a href="#">"QueryPage" on page 123</a>	CourseEnrollmentQueryPage
<a href="#">"Update" on page 145</a>	CourseEnrollmentUpdate

## Fields

[Table 459](#) details the required and read-only fields for the course enrollment object.

Table 459. Required and Read-Only Fields for the Course Enrollment Object

Field Name	Type
CourseId	Required
CandidateId	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 460](#) details the pick map fields for the course enrollment object.

Table 460. 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

Table 460. Pick Map Fields for the Course Enrollment Object

Pick Map Field	Maps To
OfferingPartnerChannelAccountManagerAlias, OfferingPartnerIntegrationId, OfferingPartnerExternalSystemId, OfferingPartnerLocation, OfferingPartnerMainPhone, OfferingPartnerPartnerName	OfferingPartnerId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 461 provides a list of the user key combinations for the course enrollment object.

Table 461. User Key Fields on the Course Enrollment Object

User Key Field Combinations
CourseId
CandidateId
EnrollDate

Table 462 details the picklists available for the course enrollment object.

Table 462. 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](#), [Team](#).

## Methods Called

[Table 463](#) details the methods called by the Deal Registration service.

Table 463. Methods Called by Deal Registration Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	DealRegistrationDelete
<a href="#">"Execute" on page 116</a>	DealRegistrationExecute
<a href="#">"Insert" on page 120</a>	DealRegistrationInsert
<a href="#">"QueryPage" on page 123</a>	DealRegistrationQueryPage
<a href="#">"Update" on page 145</a>	DealRegistrationUpdate

## Fields

[Table 464](#) details the required and read-only fields for the deal registration object.

Table 464. 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
<a href="#">Audit Fields</a>	Read-only

[Table 465](#) details the pick map fields for the deal registration object.

Table 465. Pick Map Fields for the Deal Registration Object

Pick Map Field	Maps To
PartnerExternalSystemId	PartnerId
PartnerName	PartnerId

Table 466 provides a list of the user key combinations for the deal registration object.

Table 466. User Key Fields on the Deal Registration Object

User Key Field Combinations
ExternalSystemId
DealRegistrationName and PartnerId
Id

Table 467 details the picklists available for the deal registration object.

Table 467. Picklists Available for the Deal Registration Object

Field Name
Type
Status
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](#), [Team](#).

## Methods Called

Table 468 details the methods called by the Exam service.

Table 468. Methods Called by Exam Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ExamDelete
<a href="#">"Execute" on page 116</a>	ExamExecute
<a href="#">"Insert" on page 120</a>	ExamInsert
<a href="#">"QueryPage" on page 123</a>	ExamQueryPage
<a href="#">"Update" on page 145</a>	ExamUpdate

## Fields

Table 469 details the required and read-only fields for the exam object.

Table 469. Required and Read-Only Fields for the Exam Object

Field Name	Type
Name	Required
Status	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only

Table 470 details the pick map fields for the exam object.

Table 470. Pick Map Fields for the Exam Object

Pick Map Field	Maps To
PrimaryProductProductCategory, PrimaryProductExternalSystemId, PrimaryProductDescription, PrimaryProductName, PrimaryProductPartNumber, PrimaryProductStatus, PrimaryProductIntegrationId, PrimaryProductType	PrimaryProductId
ProductCategoryName	ProductCategoryId

Table 470. Pick Map Fields for the Exam Object

Pick Map Field	Maps To
ContactEmail, ContactFirstName, ContactLastName, ContactFullName, ContactIntegrationId, ContactAccountName, ContactExternalSystemId	ContactId
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr	OwnerId

Table 471 provides a list of the user key combinations for the exam object.

Table 471. User Key Fields on the Exam Object

User Key Field Combinations
Id
Name

Table 472 details the picklists available for the exam object.

Table 472. 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 473 details the methods called by the Exam Registration service.

Table 473. Methods Called by Exam Registration Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ExamRegistrationDelete
<a href="#">"Execute" on page 116</a>	ExamRegistrationExecute
<a href="#">"Insert" on page 120</a>	ExamRegistrationInsert
<a href="#">"QueryPage" on page 123</a>	ExamRegistrationQueryPage
<a href="#">"Update" on page 145</a>	ExamRegistrationUpdate

## Fields

Table 474 details the required and read-only fields for the exam registration object.

Table 474. Required and Read-Only Fields for the Exam Registration Object

Field Name	Type
ExamId	Required
CandidateId	Required
ExamDate	Required
OwnerId	Required
<a href="#">Audit Fields</a>	Read-only

Table 475 details the pick map fields for the exam registration object.

Table 475. 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



Table 475. Pick Map Fields for the Exam Registration Object

Pick Map Field	Maps To
ObjectiveName, ObjectiveExternalSystemId	ObjectiveId
PartnerChannelAccountManagerAlias, PartnerIntegrationId, PartnerExternalSystemId, PartnerLocation, PartnerMainPhone, PartnerName,	PartnerId
OwnerFirstName, OwnerLastName, OwnerUserSignInId, OwnerAlias, OwnerFullName, OwnerIntegrationId, OwnerExternalSystemId, OwnerEMailAddr.	OwnerId

Table 476 provides a list of the user key combinations for the exam registration object.

Table 476. User Key Fields on the Exam Registration Object

User Key Field Combinations
Id
CourseId, CandidateId, OwnerId, and ExamDate

Table 477 details the picklists available for the exam registration object.

Table 477. 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, Fund Credit, Fund Debit, [MDF Request](#), [Event](#), [Objective](#), [Special Pricing Product](#).

## Methods Called

Table 478 details the methods called by the Fund service.

Table 478. Methods Called by Fund Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	FundDelete
<a href="#">"Execute" on page 116</a>	FundExecute
<a href="#">"Insert" on page 120</a>	FundInsert
<a href="#">"QueryPage" on page 123</a>	FundQueryPage
<a href="#">"Update" on page 145</a>	FundUpdate

## Fields

Table 479 details the required and read-only fields for the fund object.

Table 479. Required and Read-Only Fields for the Fund Object

Field Name	Type
FundName	Required
FundId	Required
TargetAmount	Required
<a href="#">Audit Fields</a>	Read-only

Table 480 details the pick map field for the fund object.

Table 480. Pick Map Field for the Fund Object

Pick Map Field	Maps To
Partner	PartnerId

Table 481 provides a list of the user key combinations for the fund object.

Table 481. User Key Fields on the Fund Object

User Key Field Combinations
FundId
Id

Table 482 details the picklists available for the fund object.

Table 482. 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, [Team](#).

### Methods Called

Table 483 details the methods called by the MDF Request service.

Table 483. Methods Called by MDF Request Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	MDFRequestDelete
<a href="#">"Execute" on page 116</a>	MDFRequestExecute
<a href="#">"Insert" on page 120</a>	MDFRequestInsert
<a href="#">"QueryPage" on page 123</a>	MDFRequestQueryPage
<a href="#">"Update" on page 145</a>	MDFRequestUpdate

## Fields

[Table 484](#) details the required and read-only fields for the MDF request object.

Table 484. Required and Read-Only Fields on the MDF Request Object

Field Name	Type
RequestName	Required
PrincipalPartnerAccountId	Required
FundName	Required
DueDate	Required
SubmissionStatus	Required
MDFRequestId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 485](#) details the pickmap fields for the MDF request object.

Table 485. Pick Map Fields for the MDF Request Object

Pick Map Field	Maps To
PrincipalPartnerAccountExternalSystemId	PrincipalPartnerAccountId
PrincipalPartnerAccountIntegrationId	PrincipalPartnerAccountId
PrincipalPartnerAccountPartnerName, PrincipalPartnerAccountLocation	PrincipalPartnerAccountId
PrincipalPartnerAccountChannelAccountManagerAlias	PrincipalPartnerAccountId

[Table 486](#) provides a list of the user key combinations for the MDF request object.

Table 486. User Key Fields on the MDF Request Object

User Key Field Combinations
RequestName
Id
ExternalSystemId

Table 487 details the picklists available for the MDF request object.

Table 487. 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](#), [Opportunity](#), [Partner](#), [Policy](#), [Shared Address](#), [Special Pricing Product](#), [Team](#).

### Methods Called

Table 488 details the methods called by the Partner service.

Table 488. Methods Called by Partner Service

Method	Name as Defined in Service
<a href="#">"Insert" on page 120</a>	PartnerInsert
<a href="#">"QueryPage" on page 123</a>	PartnerQueryPage
<a href="#">"Update" on page 145</a>	PartnerUpdate

## Fields

[Table 489](#) details the read-only fields for the partner object.

Table 489. Read-Only Fields on the Partner Object

Field Name	Type
PartnerName	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 490](#) details the pickmap fields for the partner object.

Table 490. Pick Map Fields for the Partner Object

Pick Map Field	Maps To
Owner	OwnerId
ParentAccountName, ParentAccountLocation	ParentAccountId
ChannelAccountManager	ChannelAccountManagerId
SourceCampaignName	SourceCampaignId
OwnerPartnerAccountPartnerName	OwnerPartnerId
PrincipalPartnerPartnerName	PrincipalPartnerId
OriginatingPartnerAccountPartnerName	OriginatingPartnerId
Territory	TerritoryId

[Table 491](#) provides a list of the user key combinations for the partner object.

Table 491. User Key Fields on the Partner Object

User Key Field Combinations
Id
ExternalSystemId
Location
PartnerName

Table 492 details the picklists available for the partner object.

Table 492. Picklists Available for the Partner Object

Field Name
CallFrequency
ComplianceStatus
Expertise
InfluenceType
MarketPotential
MarketSegment
PartnerOrgStatus
PartnerLevel
Priority
Region
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 493 details the methods called by the Price List service.

Table 493. Methods Called by Price List Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	PriceListDelete
<a href="#">"Execute" on page 116</a>	PriceListExecute
<a href="#">"Insert" on page 120</a>	PriceListInsert
<a href="#">"QueryPage" on page 123</a>	PriceListQueryPage
<a href="#">"Update" on page 145</a>	PriceListUpdate

## Fields

[Table 494](#) details the required and read-only fields for the price list object.

Table 494. Required and Read-Only Fields for the Price List Object

Field Name	Type
PriceListName	Required
EffectiveFrom	Required
Type	Required
Status	Required
CurrencyCode	Required
<a href="#">Audit Fields</a>	Read-only

There are no pick map fields for the price list object.

[Table 495](#) details the user keys for the price list object.

Table 495. User Keys for the Price List Object

Field Name
PriceListName and Type
Id
ExternalSystemId

[Table 496](#) details the picklists available for the price list object.

Table 496. 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 497](#) details the methods called by the Price List Line Item service.

Table 497. Methods Called by Price List Line Item Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	PriceListLineItemDelete
<a href="#">"Execute" on page 116</a>	PriceListLineItemExecute
<a href="#">"Insert" on page 120</a>	PriceListLineItemInsert
<a href="#">"QueryPage" on page 123</a>	PriceListLineItemQueryPage
<a href="#">"Update" on page 145</a>	PriceListLineItemUpdate

## Fields

[Table 498](#) details the required and read-only fields for the price list line item object.

Table 498. Required and Read-Only Fields for the Price List Line Item Object

Field Name	Type
PriceListId	Required
ProductId	Required
Type	Required
<a href="#">Audit Fields</a>	Read-only

[Table 499](#) details the pick map field for the price list line item object.

Table 499. 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 500 details the user keys for the price list line item object.

Table 500. User Keys for the Price List Line Item Object

Field Name
Id
ExternalSystemId
ProductId and PriceListId and PriceType

Table 501 details the picklists available for the price list line item object.

Table 501. 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 502 details the methods called by the Special Pricing Product service.

Table 502. Methods Called by Special Pricing Product Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SPRequestLineItemDelete
<a href="#">"Execute" on page 116</a>	SPRequestLineItemExecute
<a href="#">"Insert" on page 120</a>	SPRequestLineItemInsert
<a href="#">"QueryPage" on page 123</a>	SPRequestLineItemQueryPage
<a href="#">"Update" on page 145</a>	SPRequestLineItemUpdate

## Fields

[Table 503](#) details the required and read-only fields for the special pricing product object.

Table 503. Required and Read-Only Fields on the Special Pricing Product Object

Field Name	Type
ProductId	Required
SPRequestId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 504](#) details the pickmap fields for the special pricing product object.

Table 504. Pick Map Fields for the Special Pricing Product Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
ProductName, ProductCategory	ProductId

[Table 505](#) provides a list of the user key combinations for the special pricing product object.

Table 505. User Key Fields on the Special Pricing Product Object

User Key Field Combinations
Id
ExternalSystemId
ItemNumber

## 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](#), [Team](#).

## Methods Called

Table 506 details the methods called by the SP Request service.

Table 506. Methods Called by SP Request Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SPRequestDelete
<a href="#">"Execute" on page 116</a>	SPRequestExecute
<a href="#">"Insert" on page 120</a>	SPRequestInsert
<a href="#">"QueryPage" on page 123</a>	SPRequestQueryPage
<a href="#">"Update" on page 145</a>	SPRequestUpdate

## Fields

Table 507 details the required and read-only fields for the SP request object.

Table 507. Read-Only Fields on the SP Request Object

Field Name	Type
SPRequestName	Required
PrincipalPartnerAccountId	Required
SubmissionStatus	Required
SPRequestId	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 508 details the pickmap fields for the SP request object.

Table 508. Pick Map Fields for the SP Request Object

Pick Map Field	Maps To
PrincipalPartnerAccountExternalSystemId	PrincipalPartnerAccountId
PrincipalPartnerAccountIntegrationId	PrincipalPartnerAccountId
PrincipalPartnerAccountPartnerName, PrincipalPartnerAccountLocation	PrincipalPartnerAccountId
PrincipalPartnerAccountChannelAccountManagerAlias	PrincipalPartnerAccountId

Table 509 provides a list of the user key combinations for the SP request object.

Table 509. User Key Fields on the SP Request Object

User Key Field Combinations
Id
ExternalSystemId
SPRequestName
FundId and RequestDate

Table 510 details the picklists available for the SP request object.

Table 510. 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 615](#)
- [“Blocked Product” on page 616](#)
- [“Business Plan” on page 618](#)
- [“Call Activity History” on page 620](#)
- [“Call Product Detail” on page 621](#)
- [“Contact Best Times” on page 623](#)

- "Contact State License" on page 624
- "Event" on page 626
- "HCP Contact Allocation" on page 628
- "Inventory Audit Report" on page 630
- "Inventory Period" on page 632
- "Message Response" on page 633
- "Messaging Plan" on page 635
- "Messaging Plan Item" on page 637
- "Messaging Plan Item Relation" on page 639
- "Objective" on page 641
- "Order" on page 643
- "Order Item" on page 645
- "Plan Account" on page 646
- "Plan Contact" on page 648
- "Plan Opportunity" on page 650
- "Product Indication" on page 651
- "Related Disclaimer" on page 652
- "Sample Disclaimer" on page 654
- "Sample Inventory" on page 657
- "Sample Lot" on page 659
- "Sample Request" on page 661
- "Sample Request Item" on page 663
- "Sample Transaction" on page 664
- "Signature" on page 667
- "Transaction Item" on page 668

**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

[Book](#)

### Methods Called

[Table 511](#) details the methods called by the Allocation service.

Table 511. Methods Called by Allocation Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	AllocationDelete
<a href="#">"Execute" on page 116</a>	AllocationExecute
<a href="#">"Insert" on page 120</a>	AllocationInsert
<a href="#">"QueryPage" on page 123</a>	AllocationQueryPage
<a href="#">"Update" on page 145</a>	AllocationUpdate

### Fields

[Table 512](#) details the required and read-only fields for the allocation object.

Table 512. Required and Read-Only Fields for the Allocation Object

Field Name	Type
AllocationType	Required
OwnerAlias	Required
ProductName	Required
StartDate	Required
ActivePeriod	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 513 details the pick map fields for the allocation object.

Table 513. 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 514 provides a list of the user key combinations for the allocation object.

Table 514. User Key Fields on the Allocation Object

User Key Field Combinations
Id
ExternalSystemId
OwnerId and ProductId and StartDate and EndDate and AllocationType

Table 515 details the picklists available for the allocation object.

Table 515. Picklists Available for the Allocation Object

Field Name
AllocationType

## 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 516](#) details the methods called by the Blocked Product service.

Table 516. Methods Called by Blocked Product Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	BlockedProductDelete
<a href="#">"Execute" on page 116</a>	BlockedProductExecute
<a href="#">"Insert" on page 120</a>	BlockedProductInsert
<a href="#">"QueryPage" on page 123</a>	BlockedProductQueryPage
<a href="#">"Update" on page 145</a>	BlockedProductUpdate

## Fields

[Table 517](#) details the required and read-only fields for the blocked product object.

Table 517. Required and Read-Only Fields for the Blocked Product Object

Field Name	Type
ProductId	Required
ContactId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 518](#) details the pick map fields for the blocked product object.

Table 518. 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 519 provides a list of the user key combinations for the blocked product object.

Table 519. User Key Fields on the Blocked Product Object

User Key Field Combinations
Id
ExternalSystemId
ProductId and ContactId

## 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](#), [Book](#), [Business Plan](#), [Event](#), [Messaging Plan](#), [Objective](#), [Plan Account](#), [Plan Contact](#), [Plan Opportunity](#), [Sales Assessment](#), [Sales Assessment Value](#), [Team](#).

### Methods Called

Table 520 details the methods called by the Business Plan service.

Table 520. Methods Called by Business Plan Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	BusinessPlanDelete
<a href="#">"Execute" on page 116</a>	BusinessPlanExecute
<a href="#">"Insert" on page 120</a>	BusinessPlanInsert
<a href="#">"QueryPage" on page 123</a>	BusinessPlanQueryPage
<a href="#">"Update" on page 145</a>	BusinessPlanUpdate

## Fields

[Table 521](#) details the required and read-only fields for the business plan object.

Table 521. Required and Read-Only Fields for the Business plan Object

Field Name	Type
PlanName	Required
PeriodName	Required
Type	Required
Status	Required
<a href="#">Audit Fields</a>	Read-only

[Table 522](#) details the pick map fields for the business plan object.

Table 522. 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 523](#) provides a list of the user key combinations for the business plan object.

Table 523. User Key Fields on the Business plan Object

User Key Field Combinations
Id

Table 523. User Key Fields on the Business plan Object

User Key Field Combinations
ExternalSystemId
PeriodId and PlanName

[Table 524](#) details the picklists available for the business plan object.

Table 524. Picklists Available for the Business plan Object

Field Name
Type
Status

## Call Activity History

The call activity history object stores data that enables sales managers and sales representatives to analyze call activity.

### Usage

The call activity history is generated only if the Generate Call Activity History setting is selected in the Oracle CRM On Demand UI. This setting is available with the Oracle CRM On Demand Life Sciences Edition only.

The fields returned by the CallActivityHistoryQueryPage method includes a set of fields used to describe and track an activity record itself.

### Child Objects

None.

### Methods Called

[Table 525](#) details the methods called by the Call Activity History service.

Table 525. Methods Called by Call Activity History Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	CallActivityHistoryQueryPage

## Fields

No pickmaps are available for the call activity history object

[Table 526](#) provides a list of the user key combinations for the call activity history object.

Table 526. User Key Fields on the Call Activity History Object

User Key Field Combinations
ActivityId
Type
ProductId
LotName

No picklists are available for the call activity history object.

## Call Product Detail

The call product detail object stores product detail information associated with sales calls.

## Child Objects

None.

## Methods Called

[Table 527](#) details the methods called by the Call Product Detail service.

Table 527. Methods Called by Call Product Detail Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ProductsDetailedDelete
<a href="#">"Execute" on page 116</a>	ProductsDetailedExecute
<a href="#">"Insert" on page 120</a>	ProductsDetailedInsert
<a href="#">"QueryPage" on page 123</a>	ProductsDetailedQueryPage
<a href="#">"Update" on page 145</a>	ProductsDetailedUpdate

## Fields

[Table 528](#) details the required and read-only fields for the call product detail object.

Table 528. Required and Read-Only Fields for the Call Product Detail Object

Field Name	Type
Product	Required
<a href="#">Audit Fields</a>	Read-only

[Table 529](#) details the pick map fields for the call product detail object.

Table 529. Pick Map Fields for the Call Product Detail Object

Pick Map Field	Maps To
ExternalSystemId	ProductId
IntegrationId	ProductId

[Table 530](#) provides a list of the user key combinations for the call product detail object.

Table 530. User Key Fields on the Call Product Detail Object

User Key Field Combinations
ProductId
IntegrationId
ExternalSystemId

[Table 531](#) details the picklists available for the call product detail object.

Table 531. Picklists Available for the Call Product Detail Object

Field Name
Indication
Issues
Type

## Contact Best Times

The contact best times 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 times 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 532 details the methods called by the Contact Best Time service.

Table 532. Methods Called by Contact Best Time Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ContactBestTimesDelete
<a href="#">"Execute" on page 116</a>	ContactBestTimesExecute
<a href="#">"Insert" on page 120</a>	ContactBestTimesInsert
<a href="#">"QueryPage" on page 123</a>	ContactBestTimesQueryPage
<a href="#">"Update" on page 145</a>	ContactBestTimesUpdate

### Fields

Table 533 details the required and read-only fields for the contact best times object.

Table 533. Required and Read-Only Fields for the Contact Best Times Object

Field Name	Type
StartDay	Required
StartTime	Required
EndTime	Required
FullAddress	Required
ContactFullName	Required
<a href="#">Audit Fields</a>	Read-only

Table 534 details the pick map fields for the contact best times object.

Table 534. Pick Map Fields for the Contact Best Times Object

Pick Map Field	Maps To
AddressExternalSystemId	AddressId
AddressIntegrationId	AddressId
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId

Table 535 provides a list of the user key combinations for the contact best times object.

Table 535. User Key Fields on the Contact Best Times Object

User Key Field Combinations
ContactId
AddressId
ExternalSystemId

Table 536 details the picklists available for the contact best times object.

Table 536. Picklists Available for the Contact Best Times 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 537](#) details the methods called by the contact state license service.

Table 537. Methods Called by Contact State License Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ContactLicenseDelete
<a href="#">"Execute" on page 116</a>	ContactLicenseExecute
<a href="#">"Insert" on page 120</a>	ContactLicenseInsert
<a href="#">"QueryPage" on page 123</a>	ContactLicenseQueryPage
<a href="#">"Update" on page 145</a>	ContactLicenseUpdate

## Fields

[Table 538](#) details the required and read-only fields for the contact state license object.

Table 538. Required and Read-Only Fields for the Contact State License Object

Field Name	Type
ContactFullName	Required
LicenseNumber	Required
<a href="#">Audit Fields</a>	Read-only

[Table 539](#) details the pick map fields for the contact state license object.

Table 539. 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 540 provides a list of the user key combinations for the contact state license object.

Table 540. User Key Fields on the Contact State License Object

User Key Field Combinations
Id
ExternalSystemId
ContactId and State
LicenseNumber

Table 541 details the picklists available for the contact state license object.

Table 541. Picklists Available for the Contact State License Object

Field Name
State
Status

## Event

The Event object allows you to plan and track events such as a medical educational seminar, a symposium, or a customer appreciation event. An event can be as simple as a lunch-and-learn session in a physician's office or as complex as a seminar series or national sales meeting.

**NOTE:** The Event Web service was labelled as the MedEvent Web service before Oracle CRM On Demand Release 25.

## Child Objects

[Account](#), [Attachment](#), [Book](#), [Business Plan](#), [Campaign](#), [Contact](#), [Custom Object 4 through 15](#), [Fund](#), [Invitee](#), [Lead](#), [Objective](#), [Opportunity](#), [Team](#)

## Methods Called

Table 542 details the methods called by the Event service.

Table 542. Methods Called by Event Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	MedEdDelete
<a href="#">"Execute" on page 116</a>	MedEdExecute

Table 542. Methods Called by Event Service

Method	Name as Defined in Service
<a href="#">"Insert" on page 120</a>	MedEdInsert
<a href="#">"QueryPage" on page 123</a>	MedEdQueryPage
<a href="#">"Update" on page 145</a>	MedEdUpdate

## Fields

[Table 543](#) details the required and read-only fields for the Event object.

Table 543. Required and Read-Only Fields on the Event Object

Field Name	Type
EndDate	Required
Name	Required
Objective	Required
StartDate	Required
OwnerFullName	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 544](#) details the pickmap fields for the Event object.

Table 544. Pick Map Fields for the Event Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

[Table 545](#) provides a list of the user key combinations for the Event object.

Table 545. User Key Fields on the Event Object

User Key Field Combinations
Id
ExternalSystemId
Name

Table 546 details the picklists available for the Event object.

Table 546. Picklists Available for the Event Object

Field Name
Status
Type

## HCP Contact Allocation

The HCP contact allocation object stores information about the total amount of a product or sample that a health care professional (HCP) contact, (for example, a physician) can receive from sales representatives.

### Child Objects

[Book](#)

### Methods Called

Table 547 details the methods called by the HCP contact allocation service.

Table 547. Methods Called by HCP Contact Allocation Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	HCPContactAllocationDelete
<a href="#">"Execute" on page 116</a>	HCPContactAllocationExecute
<a href="#">"Insert" on page 120</a>	HCPContactAllocationInsert
<a href="#">"QueryPage" on page 123</a>	HCPContactAllocationQueryPage
<a href="#">"Update" on page 145</a>	HCPContactAllocationUpdate

### Fields

Table 548 details the required and read-only fields for the HCP contact allocation object.

Table 548. Required and Read-Only Fields for the HCP Contact Allocation Object

Field Name	Type
Type	Required
StartDate	Required
ContactId	Required

Table 548. Required and Read-Only Fields for the HCP Contact Allocation Object

Field Name	Type
ProductName	Either one of these fields is required
ProductId	
<a href="#">Audit Fields</a>	Read-only

[Table 549](#) details the pick map fields for the HCP contact allocation object.

Table 549. Pick Map Fields for the HCP Contact Allocation License Object

Pick Map Field	Maps To
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignInId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 550](#) provides a list of the user key combinations for the HCP contact allocation object.

Table 550. User Key Fields on the HCP Contact Allocation Object

User Key Field Combinations
Id
ExternalSystemId

[Table 551](#) details the picklists available for the HCP contact allocation object.

Table 551. Picklists Available for the HCP Contact Allocation Object

Field Name
Type

## 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.

### Support for Attachments

You can exchange attachments through the following fields:

UI Display Name	Integration Tag Name
Attachment	AttachFileFullName
Attachment: Extension	AttachFileExt
Attachment: File Name	AttachFileName
Attachment: Size (in Bytes)	AttachFileSizeDisp

The Attachment tag is used for the actual attachment and is a string type Base64.

### Child Objects

[Book](#), [Sample Inventory](#)

### Methods Called

[Table 552](#) details the methods called by the Inventory Audit Report service.

Table 552. Methods Called by Inventory Audit Report Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	InventoryAuditReportDelete
<a href="#">"Execute" on page 116</a>	InventoryAuditReportExecute
<a href="#">"Insert" on page 120</a>	InventoryAuditReportInsert
<a href="#">"QueryPage" on page 123</a>	InventoryPeriodQueryPage
<a href="#">"Update" on page 145</a>	InventoryPeriodUpdate

## Fields

[Table 553](#) details the required and read-only fields for the inventory audit report object.

Table 553. Required and Read-Only Fields for the Inventory Audit Report Object

Field Name	Type
Type	Required
Status	Required
Reason	Required
CountCompletedOn	Required
InventoryPeriodStartDate	Required
OwnerUserSignId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 554](#) details the pick map fields for the inventory audit report object.

Table 554. 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 555](#) provides a list of the user key combinations for the inventory audit report object.

Table 555. User Key Fields on the Inventory Audit Report Object

User Key Field Combinations
Id
ExternalSystemId
InventoryPeriodId and OwnerId and CountCompletedOn
Type

Table 556 details the picklists available for the inventory audit report object.

Table 556. 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

[Book](#), [Inventory Audit Report](#), [Sample Inventory](#), [Sample Transaction](#), [Sample Transaction](#).

### Methods Called

Table 557 details the methods called by the Inventory Period service.

Table 557. Methods Called by Inventory Period Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	InventoryPeriodDelete
<a href="#">"Execute" on page 116</a>	InventoryPeriodExecute
<a href="#">"Insert" on page 120</a>	InventoryPeriodInsert
<a href="#">"QueryPage" on page 123</a>	InventoryPeriodQueryPage
<a href="#">"Update" on page 145</a>	InventoryPeriodUpdate

### Fields

Table 558 details the required and read-only fields for the inventory period object.

Table 558. Required and Read-Only Fields for the Inventory Period Object

Field Name	Type
StartDate	Required
OwnerId	Required



Table 558. Required and Read-Only Fields for the Inventory Period Object

Field Name	Type
ActiveFlg	Required
ReconciledFlg	Required
<a href="#">Audit Fields</a>	Read-only

[Table 559](#) details the pick map fields for the inventory period object.

Table 559. Pick Map Field for the Inventory Period Object

Pick Map Field	Maps To
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 560](#) provides a list of the user key combinations for the inventory period object.

Table 560. User Key Fields on the Inventory Period Object

User Key Field Combinations
Id
ExternalSystemId
OwnerId and StartDate

## 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

[Custom Object 1 through 3](#), [Custom Object 4 through 15](#)

## Methods Called

[Table 561](#) details the methods called by the Message Response service.

Table 561. Methods Called by Message Response Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	MessageResponseDelete
<a href="#">"Execute" on page 116</a>	MessageResponseExecute
<a href="#">"Insert" on page 120</a>	MessageResponseInsert
<a href="#">"QueryPage" on page 123</a>	MessageResponseQueryPage
<a href="#">"Update" on page 145</a>	MessageResponseUpdate

## Fields

[Table 562](#) details the required and read-only fields for the message response object.

Table 562. Required and Read-Only Fields on the Message Response Object

Field Name	Type
Message	Required
SequenceNumber	Required
Duration	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 563](#) details the pickmap fields for the message response object.

Table 563. 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

Table 563. Pick Map Fields for the Message Response Object

Pick Map Field	Maps To
PlanName	PlanId
PlanExternalSystemId	PlanId
PlanIntegrationId	PlanId
PlanItemName	PlanItemId
PlanItemExternalSystemId	PlanItemId
PlanItemIntegrationId	PlanItemId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 564 provides a list of the user key combinations for the message response object.

Table 564. User Key Fields on the Message Response Object

User Key Field Combinations
Id
ExternalSystemId
PlanName and SolutionName

Table 565 details the picklists available for the message response object.

Table 565. 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](#), [Business Plan](#), [Custom Object 1 through 3](#), [Custom Object 4 through 15](#), [Message Response](#), [Messaging Plan Item](#).

## Methods Called

[Table 566](#) details the methods called by the Messaging Plan service.

Table 566. Methods Called by Messaging Plan Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	MessagePlanDelete
<a href="#">"Execute" on page 116</a>	MessagePlanExecute
<a href="#">"Insert" on page 120</a>	MessagePlanInsert
<a href="#">"QueryPage" on page 123</a>	MessagePlanQueryPage
<a href="#">"Update" on page 145</a>	MessagePlanUpdate

## Fields

[Table 567](#) details the required and read-only fields for the messaging plan object.

Table 567. Required and 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
<a href="#">Audit Fields</a>	Read-only

[Table 568](#) details the pickmap fields for the messaging plan object.

Table 568. Pick Map Fields for the Messaging Plan Object

Pick Map Field	Maps To
SolutionTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId

Table 568. Pick Map Fields for the Messaging Plan Object

Pick Map Field	Maps To
EffectivePeriodName	EffectivePeriodId
EffectivePeriodExternalSystemId	EffectivePeriodId
EffectivePeriodIntegrationId	EffectivePeriodId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 569 provides a list of the user key combinations for the messaging plan object.

Table 569. User Key Fields on the Messaging Plan Object

User Key Field Combinations
Id
ExternalSystemId
Name and OwnerId

Table 570 details the picklists available for the messaging plan object.

Table 570. 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

Custom Object 1 through 3, Custom Object 4 through 15, Messaging Plan Item Relation, Message Response.

## Methods Called

Table 571 details the methods called by the Messaging Plan Item service.

Table 571. Methods Called by Messaging Plan Item Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	MsgPlanItemDelete
<a href="#">"Execute" on page 116</a>	MsgPlanItemExecute
<a href="#">"Insert" on page 120</a>	MsgPlanItemInsert
<a href="#">"QueryPage" on page 123</a>	MsgPlanItemQueryPage
<a href="#">"Update" on page 145</a>	MsgPlanItemUpdate

## Fields

Table 572 details the required and read-only fields for the messaging plan item object.

Table 572. Required and Read-Only Fields on the Messaging Plan Item Object

Field Name	Type
SequenceNumber	Required
DisclosureMessage	Required
Type	Required
ParentMessagePlanName	Required
<a href="#">Audit Fields</a>	Read-only

Table 573 details the pickmap fields for the messaging plan item object.

Table 573. Pick Map Fields for the Messaging Plan Item Object

Pick Map Field	Maps To
ParentMessagePlanName	ParentMessagePlanId
ParentMessagePlanExternalSystemId	ParentMessagePlanId
ParentMessagePlanIntegrationId	ParentMessagePlanId
SolutionNameTitle	SolutionId

Table 573. Pick Map Fields for the Messaging Plan Item Object

Pick Map Field	Maps To
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 574 provides a list of the user key combinations for the messaging plan item object.

Table 574. User Key Fields on the Messaging Plan Item Object

User Key Field Combinations
Id
ExternalSystemId
ParentMessagePlanId and SolutionNameId and SequenceNumber and Type

Table 575 details the picklists available for the messaging plan item object.

Table 575. Picklists Available for the Messaging Plan Item Object

Field Name
Type
Status
Rating
SendType

## 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

[Custom Object 1 through 3](#), [Custom Object 4 through 15](#)

## Methods Called

[Table 576](#) details the methods called by the Messaging Plan Item Relation service.

Table 576. Methods Called by Messaging Plan Item Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	MsgPlanItemRelationDelete
<a href="#">"Execute" on page 116</a>	MsgPlanItemRelationExecute
<a href="#">"Insert" on page 120</a>	MsgPlanItemRelationInsert
<a href="#">"QueryPage" on page 123</a>	MsgPlanItemRelationQueryPage
<a href="#">"Update" on page 145</a>	MsgPlanItemRelationUpdate

## Fields

[Table 577](#) details the required and read-only fields for the messaging plan item relation object.

Table 577. Required and Read-Only Fields on the Messaging Plan Item Relation Object

Field Name	Type
Type	Required
ParentMPIItemName	Required
<a href="#">Audit Fields</a>	Read-only

[Table 578](#) details the pickmap fields for the messaging plan item relation object.

Table 578. Pick Map Fields for the Messaging Plan Item Relation Object

Pick Map Field	Maps To
ParentMPIItemName	ParentMPIItemId
ParentMPIItemExternalSystemId	ParentMPIItemId
ParentMPIItemIntegrationId	ParentMPIItemid
SolutionNameTitle	SolutionId
SolutionExternalSystemId	SolutionId
SolutionIntegrationId	SolutionId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId



Table 579 provides a list of the user key combinations for the messaging plan item relation object.

Table 579. User Key Fields on the Messaging Plan Item Relation Object

User Key Field Combinations
Id
ExternalSystemId
ParentMPIId and SolutionNameId and Type

Table 580 details the picklists available for the messaging plan item relation object.

Table 580. Picklists Available for the Messaging Plan Item Relation Object

Field Name
Type
Status
Rating

## 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](#), [Book](#), [Objective](#), [Opportunity](#), [Sales Assessment](#), [Sales Assessment Value](#), [Team](#).

## Methods Called

Table 581 details the methods called by the objective service.

Table 581. Methods Called by Objective Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ObjectiveDelete
<a href="#">"Execute" on page 116</a>	ObjectiveExecute
<a href="#">"Insert" on page 120</a>	ObjectiveInsert
<a href="#">"QueryPage" on page 123</a>	ObjectiveQueryPage
<a href="#">"Update" on page 145</a>	ObjectiveUpdate

## Fields

[Table 582](#) details the required and read-only fields for the objective object.

Table 582. Required and Read-Only Fields for the Objective Object

Field Name	Type
ObjectiveName	Required
Type	Required
Status	Required
PeriodName	Required
<a href="#">Audit Fields</a>	Read-only

[Table 583](#) details the pick map fields for the objective object.

Table 583. Pick Map Fields for the Objective Object

Pick Map Field	Maps To
ParentObjectiveObjectiveName	ParentObjectiveId
ParentObjectiveExternalSystemId	ParentObjectiveId
ParentObjectiveIntegrationId	ParentObjectiveId
PlanNamePlanName	PlanNameId
PlanNameExternalSystemId	PlanNameId
PlanNameIntegrationId	PlanNameId
AccountNameAccountName	AccountId
AccountExternalSystemId	AccountId
AccountIntegrationId	AccountId
ContactNameContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
ProductNameName	ProductId
ProductdExternalSystemId	ProductId
ProductIntegrationId	ProductId
PeriodName	PeriodId
PeriodExternalSystemId	PeriodId
PeriodIntegrationId	PeriodId

Table 583. Pick Map Fields for the Objective Object

Pick Map Field	Maps To
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 584 provides a list of the user key combinations for the objective object.

Table 584. User Key Fields on the Objective Object

User Key Field Combinations
Id
ExternalSystemId
AccountNameId and ContactNameId and ProductNameId and PeriodId
ObjectiveName

Table 585 details the picklists available for the objective object.

Table 585. 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.

For Insert requests, if the Type field is not specified as empty or as a picklist value, then it defaults to the value Order.

## Child Objects

[Book](#), [Order Item](#), [Team](#)

## Methods Called

[Table 586](#) details the methods called by the Order service.

Table 586. Methods Called by Order Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	OrderDelete
<a href="#">"Execute" on page 116</a>	OrderExecute
<a href="#">"Insert" on page 120</a>	OrderInsert
<a href="#">"QueryPage" on page 123</a>	OrderQueryPage
<a href="#">"Update" on page 145</a>	OrderUpdate

## Fields

[Table 587](#) details the required and read-only fields for the order object.

Table 587. 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
<a href="#">Audit Fields</a>	Read-only

[Table 588](#) details the pick map fields for the order object.

Table 588. Pick Map Fields for the Order Object

Pick Map Field	Maps To
ContactFullName	ContactId

Table 589 provides a list of the user key combinations for the order object.

Table 589. User Key Fields on the Order Object

User Key Field Combinations
Id
OrderNumber

Table 590 details the picklists for the order object.

Table 590. Picklist Fields on the Order Object

Field Name
Type

## 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 591 details the methods called by the Order Item service.

Table 591. Methods Called by Order Item Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	OrderItemDelete
<a href="#">"Execute" on page 116</a>	OrderItemExecute
<a href="#">"Insert" on page 120</a>	OrderItemInsert
<a href="#">"QueryPage" on page 123</a>	OrderItemQueryPage
<a href="#">"Update" on page 145</a>	OrderItemUpdate

## Fields

[Table 592](#) details the required and read-only fields for the order item object.

Table 592. Required and Read-Only Fields for the Order Item Object

Field Name	Type
OrderItemNum	Required
Quantity	Required
ProductName	Required
OrderNumber	Required
OrderCreationStatus	Required
<a href="#">Audit Fields</a>	Read-only

[Table 593](#) details the pick map fields for the order item object.

Table 593. Pick Map Fields for the Order Item Object

Pick Map Field	Maps To
ProductName	ProductId
OrderNumber	OrderId

[Table 594](#) provides a list of the user key combinations for the order item object.

Table 594. 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 595](#) details the methods called by the plan account service.

Table 595. Methods Called by Plan Account Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	AccountBusinessPlanDelete
<a href="#">"Execute" on page 116</a>	AccountBusinessPlanExecute
<a href="#">"Insert" on page 120</a>	AccountBusinessPlanInsert
<a href="#">"QueryPage" on page 123</a>	AccountBusinessPlanQueryPage
<a href="#">"Update" on page 145</a>	AccountBusinessPlanUpdate

## Fields

[Table 596](#) details the required and read-only fields for the plan account object.

Table 596. Required and Read-Only Fields for the Plan Account Object

Field Name	Type
AccountIDAccountName	Required
BusinessPlanPlanName	Required
ContactId	Required
RelationshipCode	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 597](#) details the pick map fields for the plan account object.

Table 597. 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

Table 597. Pick Map Fields for the Plan Account Object

Pick Map Field	Maps To
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 598 provides a list of the user key combinations for the plan account object.

Table 598. 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 599 details the methods called by the plan contact service.

Table 599. Methods Called by Plan Contact Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ContactBusinessPlanDelete
<a href="#">"Execute" on page 116</a>	ContactBusinessPlanExecute
<a href="#">"Insert" on page 120</a>	ContactBusinessPlanInsert
<a href="#">"QueryPage" on page 123</a>	ContactBusinessPlanQueryPage
<a href="#">"Update" on page 145</a>	ContactBusinessPlanUpdate



## Fields

[Table 600](#) details the required and read-only fields for the plan contact object.

Table 600. Required and Read-Only Fields for the Plan Contact Object

Field Name	Type
ContactId	Required
BusinessPlanPlanName	Required
<a href="#">Audit Fields</a>	Read-only

[Table 601](#) details the pick map fields for the plan contact object.

Table 601. 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 602](#) provides a list of the user key combinations for the plan contact object.

Table 602. 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 603](#) details the methods called by the Plan Opportunity service.

Table 603. Methods Called by Plan Opportunity Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	PlanOpportunityDelete
<a href="#">"Execute" on page 116</a>	PlanOpportunityExecute
<a href="#">"Insert" on page 120</a>	PlanOpportunityInsert
<a href="#">"QueryPage" on page 123</a>	PlanOpportunityQueryPage
<a href="#">"Update" on page 145</a>	PlanOpportunityUpdate

### Fields

[Table 604](#) details the required and read-only fields for the plan opportunity object.

Table 604. Required and Read-Only Fields for the Plan Opportunity Object

Field Name	Type
OpportunityName	Required
PlanName	Required
<a href="#">Audit Fields</a>	Read-only

[Table 605](#) details the pick map fields for the plan opportunity object.

Table 605. Pick Map Fields for the Plan Opportunity Object

Pick Map Field	Maps To
OpportunityName	OpportunityId
OpportunityExternalSystemId	OpportunityId
OpportunityIntegrationId	OpportunityId

Table 605. Pick Map Fields for the Plan Opportunity Object

Pick Map Field	Maps To
PlanName	PlanId
PlanExternalSystemId	PlanId
PlanIntegrationId	PlanId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 606 provides a list of the user key combinations for the plan opportunity object.

Table 606. User Key Fields on the Plan Opportunity Object

User Key Field Combinations
Id
ExternalSystemId
PlanId and OpportunityId
OpportunityName

## Product Indication

The product indication object stores the combination of a symptom or malady and the specific product or drug that alleviates the symptom or malady that is typically detailed by a pharmaceutical sales representative during a call with a health practitioner.

### Child Objects

None

### Methods Called

Table 607 details the methods called by the Product Indication service.

Table 607. Methods Called by Product Indication Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ProductIndicationDelete
<a href="#">"Execute" on page 116</a>	ProductIndicationExecute
<a href="#">"Insert" on page 120</a>	ProductIndicationInsert

Table 607. Methods Called by Product Indication Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	ProductIndicationQueryPage
<a href="#">"Update" on page 145</a>	ProductIndicationUpdate

## Fields

[Table 608](#) details the required and read-only fields for the product indication object.

Table 608. Required and Read-Only Fields for the Product Indication Object

Field Name	Type
IndicationName	Required
ProductName	Required
<a href="#">Audit Fields</a>	Read-only

[Table 609](#) details the pick map fields for the product indication object.

Table 609. Pick Map Fields for the Product Indication Object

Pick Map Field	Maps To
ProductName	ProductId

[Table 610](#) details the user keys for the product indication object.

Table 610. User Keys for the Product Indication Object

Field Name
ProductName
IndicationName

## Related Disclaimer

The related disclaimer object stores the sample disclaimer text for a particular language for a multilingual sample disclaimer.

## Child Objects

None

## Usage

A number of restrictions apply to the usage of the Related Disclaimer service:

- Related disclaimer child records can be inserted and associated only with an inactive sample disclaimer parent, and cannot be inserted and associated with active or expired sample disclaimers.
- Related disclaimer child records can be deleted only from an inactive sample disclaimer parent, and cannot be deleted from active or expired sample disclaimers.
- A related disclaimer child cannot exist without a parent sample disclaimer.
- All record locking rules must be honored by the Related Disclaimer service.

**NOTE:** You might find that some of these restrictions do not apply for your company depending on whether your company administrator has set up the appropriate record locking rules using the process administration functionality of Oracle CRM On Demand. For more information about process administration, see *Oracle CRM On Demand Online Help*.

## Methods Called

Table 611 details the methods called by the Related Disclaimer service.

Table 611. Methods Called by Related Disclaimer Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	RelatedDisclaimerDelete
<a href="#">"Execute" on page 116</a>	RelatedDisclaimerExecute
<a href="#">"Insert" on page 120</a>	RelatedDisclaimerInsert
<a href="#">"QueryPage" on page 123</a>	RelatedDisclaimerQueryPage
<a href="#">"Update" on page 145</a>	RelatedDisclaimerUpdate

## Fields

Table 612 details the required and read-only fields for the related disclaimer object.

Table 612. Required and Read-Only Fields for the Related Disclaimer Object

Field Name	Type
Disclaimer	Required
Language	Required
ParentId	Required
<a href="#">Audit Fields</a>	Read-only

Table 613 details the user keys for the related disclaimer object.

Table 613. User Keys for the Related Disclaimer Object

Field Name
Id
ParentId and Language

Table 614 details the picklists available for the related disclaimer object.

Table 614. Picklists Available for the Related Disclaimer Object

Field Name
Language

## 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.

Oracle CRM On Demand supports two types of sample disclaimer:

- **Global.** A global sample disclaimer contains one disclaimer record for each company instance. This disclaimer applies to all products provided as samples, and is displayed at the time of signature capture to confirm receipt of the product samples. Only one global sample disclaimer can be active at any time. This is the default type for Oracle CRM On Demand.
- **Multilingual.** A multilingual sample disclaimer contains an unlimited number of related disclaimers text records in multiple languages, grouped by the countries that apply to the disclaimer. Each sample disclaimer record consists of a header section, and a language-specific Related Disclaimer section that contains the translated disclaimer text required by the company. Only one set of multilingual sample disclaimers can be active within a country at any time, however more than one set of multilingual sample disclaimers can be active at the same time spanning multiple countries. Before you can use multilingual sample disclaimers, your company administrator must set up special page layouts as described in *Configuration Guide for Oracle CRM On Demand Life Sciences Edition*.

For more information about sample disclaimers, see *Oracle CRM On Demand Online Help*.

## Usage

A number of restrictions apply to the usage of the Sample Disclaimer service:

- A sample disclaimer can be inserted only with a Status value of Inactive. A sample disclaimer can be activated only through the UI.

- A sample disclaimer cannot have the StartDate, EndDate, or Status fields updated.
- A sample disclaimer that is active or expired cannot be updated.
- A sample disclaimer that is active or expired cannot be deleted.
- All record locking rules must be honored by the Sample Disclaimer service.

**NOTE:** You might find that some of these restrictions do not apply for your company depending on whether your company administrator has set up the appropriate record locking rules using the process administration functionality of Oracle CRM On Demand. For more information about process administration, see *Oracle CRM On Demand Online Help*.

## Child Objects

[Related Disclaimer](#)

## Methods Called

[Table 615](#) details the methods called by the Sample Disclaimer service.

Table 615. Methods Called by Sample Disclaimer Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SignatureDisclaimerDelete
<a href="#">"Execute" on page 116</a>	SignatureDisclaimerExecute
<a href="#">"Insert" on page 120</a>	SignatureDisclaimerInsert
<a href="#">"QueryPage" on page 123</a>	SignatureDisclaimerQueryPage
<a href="#">"Update" on page 145</a>	SignatureDisclaimerUpdate

## Fields

[Table 616](#) details the required and read-only fields for the sample disclaimer object.

Table 616. Required and Read-Only Fields for the Sample Disclaimer Object

Field Name	Type
DisclaimerText	Required (global disclaimers only)
Status	Required (for inserts)
Country	Required (multilingual disclaimers only)
Language	Required (multilingual disclaimers only)
<a href="#">Audit Fields</a>	Read-only

**NOTE:** The DisclaimerText field is required for global disclaimers only. You can create a sample disclaimer without disclaimer text, but if you try to activate such a sample disclaimer through the UI, an error is generated. Similarly, the Country and Language fields are not required when a record is inserted, but they are required when a multilingual sample disclaimer is activated.

[Table 617](#) details the pick map field for the sample disclaimer object.

Table 617. Pick Map Field for the Sample Disclaimer Object

Pick Map Field	Maps To
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 618](#) details the user keys for the sample disclaimer object.

Table 618. User Keys for the Sample Disclaimer Object

Field Name
Id
ExternalSystemId
DisclaimerNumber



Table 619 details the picklists available for the sample disclaimer object.

Table 619. Picklists Available for the Sample Disclaimer Object

Field Name
Status
Type
Country

## 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 620 details the methods called by the Sample Inventory service.

Table 620. Methods Called by Sample Inventory Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SampleInventoryDelete
<a href="#">"Execute" on page 116</a>	SampleInventoryExecute
<a href="#">"Insert" on page 120</a>	SampleInventoryInsert
<a href="#">"QueryPage" on page 123</a>	SampleInventoryQueryPage
<a href="#">"Update" on page 145</a>	SampleInventoryUpdate

### Fields

Table 621 details the required and read-only fields for the sample inventory object.

Table 621. Required and Read-Only Fields for the Sample Inventory Object

Field Name	Type
InventoryPeriodStartDate	Required
ProductName	Required

Table 621. Required and Read-Only Fields for the Sample Inventory Object

Field Name	Type
OpeningBalance	Required
SystemCount	Read-only
Difference	Read-only
InventoryLot	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 622](#) details the pick map field for the sample inventory object.

Table 622. Pick Map Fields for the Sample Inventory Object

Pick Map Field	Maps To
LotNumLotId	LotNumId
LotNumExternalSystemId	LotNumId
LotNumIntegrationId	LotNumId
InventoryPeriodExternalSystemId	InventoryPeriodId
InventoryPeriodIntegrationId	InventoryPeriodId
AuditReportExternalSystemId	AuditReportId
AuditReportIntegrationId	AuditReportId
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 623](#) details the user keys for the sample inventory object.

Table 623. User Keys for the Sample Inventory Object

Field Name
SampleInventoryId
IntegrationId
ExternalSystemId

Table 623. User Keys for the Sample Inventory Object

Field Name
OwnerId and ProductId and InventoryPeriodId and LotNumId and AuditReportId
ProductName

## Sample Lot

The sample lot object records information about lot numbers for sample products.

### Child Objects

[Book](#), [Sample Inventory](#), [Transaction Item](#).

### Methods Called

[Table 624](#) details the methods called by the Sample Lot service.

Table 624. Methods Called by Sample Lot Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SampleLotDelete
<a href="#">"Execute" on page 116</a>	SampleLotExecute
<a href="#">"Insert" on page 120</a>	SampleLotInsert
<a href="#">"QueryPage" on page 123</a>	SampleLotQueryPage
<a href="#">"Update" on page 145</a>	SampleLotUpdate

## Fields

[Table 625](#) details the required and read-only fields for the sample lot object.

Table 625. 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
DateCutOff	Read-only
<a href="#">Audit Fields</a>	Read-only

**NOTE:** The DateCutOff and CutoffDate fields are both available. DateCutOff is a calculated field of type Date used if you require date processing in your application. CutoffDate is the text version of DateCutOff.

[Table 626](#) details the pick map field for the sample lot object.

Table 626. Pick Map Field for the Sample Lot Object

Pick Map Field	Maps To
ParentSampleNameName	ParentSampleNameId
ParentSampleNameExternalSystemId	ParentSampleNameId
ParentSampleNameIntegrationId	ParentSampleNameId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 627 details the user keys for the sample lot object.

Table 627. 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.

### Usage

For Insert requests, if the Type field is not specified as empty or as a picklist value, then it defaults to the value Sample Request.

### Child Objects

[Book](#), [Sample Request Item](#), [Team](#)

### Methods Called

Table 628 details the methods called by the Sample Request service.

Table 628. Methods Called by Sample Request Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SampleRequestDelete
<a href="#">"Execute" on page 116</a>	SampleRequestExecute
<a href="#">"Insert" on page 120</a>	SampleRequestInsert
<a href="#">"QueryPage" on page 123</a>	SampleRequestQueryPage
<a href="#">"Update" on page 145</a>	SampleRequestUpdate

## Fields

[Table 629](#) details the required and read-only fields for the sample request object.

Table 629. Required and Read-Only Fields for the Sample Request Object

Field Name	Type
ContactId	Required
ShippingAddress	Required
OrderNumber	Read-only
OrderIntegrationMessage	Read-only
OrderCreationDate	Read-only
OwnerId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 630](#) details the pick map fields for the sample request object.

Table 630. Pick Map Fields for the Sample Request Object

Pick Map Field	Maps To
OrderNumber	OrderId

[Table 631](#) details the user key combinations for the sample request object.

Table 631. User Key Fields on the Sample Request Object

User Key Field Combinations
Id
OrderNumber

[Table 632](#) details the picklists for the sample request object.

Table 632. Picklist Fields on the Sample Request Object

Field Name
OrderCreationStatus
Type

## Sample Request Item

The sample request item object tracks which products have been purchased by the customer as part of a sample request.

### Usage

In releases of Oracle CRM On Demand before Release 22, it was possible to insert a sample request item record without an associated parent sample request record. From Release 22, however, when inserting sample request item records, you must provide the Sample Request ID of the parent to provide the context for the sample request item.

Product blocking rules are respected for this object. A blocked product is a product for which your company does not want users to provide any product information, samples, or promotional items for that product to the associated contact.

Web service requests cannot insert, update, or delete records for this object if the product specified in the request is a blocked product. An error message is generated when such a request is made. For more information about blocked products, see *Oracle CRM On Demand Online Help* and [“Blocked Product” on page 616](#).

### Methods Called

[Table 633](#) details the methods called by the Sample Request Item service.

Table 633. Methods Called by Sample Request Item Service

Method	Name as Defined in Service
<a href="#">“Delete” on page 112</a>	SampleRequestItemDelete
<a href="#">“Execute” on page 116</a>	SampleRequestItemExecute
<a href="#">“Insert” on page 120</a>	SampleRequestItemInsert
<a href="#">“QueryPage” on page 123</a>	SampleRequestItemQueryPage
<a href="#">“Update” on page 145</a>	SampleRequestItemUpdate

### Fields

[Table 634](#) details the required and read-only fields for the sample request item object.

Table 634. Required and Read-Only Fields for the Sample Request Item Object

Field Name	Type
Quantity	Required
ProductName	Required
SampleRequestOrderNumber	Required

Table 634. Required and Read-Only Fields for the Sample Request Item Object

Field Name	Type
OrderItemNum	Required
<a href="#">Audit Fields</a>	Read-only

[Table 635](#) details the pick map fields for the sample request item object.

Table 635. Pick Map Fields for the Sample Request Item Object

Pick Map Field	Maps To
ProductName	ProductId

[Table 636](#) provides a list of the user key combinations for the sample request item object.

Table 636. User Key Fields on the Sample Request Item Object

User Key Field Combinations
OrderItemNum

[Table 637](#) details the picklists for the sample request item object.

Table 637. Picklist Fields on the Sample Request Item Object

Field Name
OrderCreationStatus

## 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.

## Support for Attachments

You can exchange attachments through the following fields:

UI Display Name	Integration Tag Name
Attachment	AttachFileFullName
Attachment: Extension	AttachFileExt



UI Display Name	Integration Tag Name
Attachment: File Name	AttachFileName
Attachment: Size (in Bytes)	AttachFileSizeDisp

The Attachment tag is used for the actual attachment and is a string type Base64.

## Child Objects

[Book](#), [Transaction Item](#)

## Methods Called

[Table 638](#) details the methods called by the Sample Transaction service.

Table 638. Methods Called by Sample Transaction Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	TransactionDelete
<a href="#">"Execute" on page 116</a>	TransactionExecute
<a href="#">"Insert" on page 120</a>	TransactionInsert
<a href="#">"QueryPage" on page 123</a>	TransactionQueryPage
<a href="#">"Update" on page 145</a>	TransactionUpdate

## Fields

[Table 639](#) details the required and read-only fields for the sample transaction object.

Table 639. Required and Read-Only Fields for the Sample Transaction Object

Field Name	Type
Name	Required
TransactionDate	Required
TransactionType	Required
InvoiceNumber	Required
<a href="#">Audit Fields</a>	Read-only

Table 640 details the pick map field for the sample transaction object.

Table 640. 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 641 details the user keys for the sample transaction object.

Table 641. User Keys for the Sample Transaction Object

Field Name
Id
ExternalSystemId
Name

Table 642 details the picklists available for the sample transaction object.

Table 642. 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.

### Support for Attachments

You can exchange attachments through the following fields:

UI Display Name	Integration Tag Name
Attachment	AttachFileFullName
Attachment: Extension	AttachFileExt
Attachment: File Name	AttachFileName
Attachment: Size (in Bytes)	AttachFileSizeDisp

The Attachment tag is used for the actual attachment and is a string type Base64.

### Child Objects

None

### Methods Called

Table 643 details the methods called by the Signature service.

Table 643. Methods Called by Signature Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	SignatureDelete
<a href="#">"Execute" on page 116</a>	SignatureExecute
<a href="#">"Insert" on page 120</a>	SignatureItemInsert
<a href="#">"QueryPage" on page 123</a>	SignatureQueryPage
<a href="#">"Update" on page 145</a>	SignatureUpdate

## Fields

[Table 644](#) details the required and read-only fields for the signature object.

Table 644. Required and Read-Only Fields for the Signature Object

Field Name	Type
ActivityId	Required
SignatureCtrl	Required
<a href="#">Audit Fields</a>	Read-only

[Table 645](#) details the pick map field for the signature object.

Table 645. Pick Map Field for the Signature Object

Pick Map Field	Maps To
ContactFullName	ContactId
ContactExternalSystemId	ContactId
ContactIntegrationId	ContactId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

[Table 646](#) details the user keys for the signature object.

Table 646. 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 647](#) details the methods called by the Transaction Item service.

Table 647. Methods Called by Transaction Item Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	TransactionItemDelete
<a href="#">"Execute" on page 116</a>	TransactionItemExecute
<a href="#">"Insert" on page 120</a>	TransactionItemInsert
<a href="#">"QueryPage" on page 123</a>	TransactionItemQueryPage
<a href="#">"Update" on page 145</a>	TransactionItemUpdate

## Fields

[Table 648](#) details the required and read-only fields for the transaction item object.

Table 648. Required and Read-Only Fields for the Transaction Item Object

Field Name	Type
LineNumber	Required
SampleTransactionName	Required
ProductName	Required
Quantity	Required
<a href="#">Audit Fields</a>	Read-only

[Table 649](#) details the pick map field for the transaction item object.

Table 649. 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

Table 649. Pick Map Field for the Transaction Item Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId
OwnerUserSignId	OwnerId
OwnerExternalSystemId	OwnerId
OwnerIntegrationId	OwnerId

Table 650 details the user keys for the transaction item object.

Table 650. 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 671
- "Coverage" on page 673
- "Damage" on page 675
- "Financial Account" on page 676
- "Financial Account Holder" on page 678
- "Financial Account Holding" on page 680

- ["Financial Plan" on page 681](#)
- ["Financial Product" on page 683](#)
- ["Financial Transaction" on page 684](#)
- ["Household" on page 686](#)
- ["Insurance Property" on page 688](#)
- ["Involved Party" on page 690](#)
- ["Policy" on page 691](#)
- ["Policy Holder" on page 693](#)
- ["Portfolio" on page 695](#)

## 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 651](#) details the methods called by the Claim service.

Table 651. Methods Called by Claim Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	ClaimDelete
<a href="#">"Execute" on page 116</a>	ClaimExecute
<a href="#">"Insert" on page 120</a>	ClaimInsert
<a href="#">"QueryPage" on page 123</a>	ClaimQueryPage
<a href="#">"Update" on page 145</a>	ClaimUpdate

## Fields

[Table 652](#) details the required and read-only fields for the claim object.

Table 652. Read-Only Fields on the Claim Object

Field Name	Type
ClaimNumber	Required
ReportDate	Required
LossDateandTime	Required
PolicyName	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 653](#) details the pickmap fields for the claim object.

Table 653. Pick Map Fields for the Claim Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

[Table 654](#) provides a list of the user key combinations for the claim object.

Table 654. User Key Fields on the Claim Object

User Key Field Combinations
Id
ExternalSystemId
ClaimNumber

[Table 655](#) details the picklists available for the claim object.

Table 655. Picklists Available for the Claim Object

Field Name
AbilitytoWork
AtFault
CategoryofLoss



Table 655. Picklists Available for the Claim Object

Field Name
ClassofEmployee
InjuredDuringWorkingHours
KindofLoss
LiabilitySource
LineofBusiness
LocationofLoss
LossCode
LossType
MedicalInjuryCode
PartofBodyInjured
PlaceofInjury
RelationshipptoInsured
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 656](#) details the methods called by the Coverage service.

Table 656. Methods Called by Coverage Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	CoverageDelete
<a href="#">"Execute" on page 116</a>	CoverageExecute
<a href="#">"Insert" on page 120</a>	CoverageInsert
<a href="#">"QueryPage" on page 123</a>	CoverageQueryPage
<a href="#">"Update" on page 145</a>	CoverageUpdate

## Fields

[Table 657](#) details the required and read-only fields for the coverage object.

Table 657. Required and Read-Only Fields on the Coverage Object

Field Name	Type
CoverageName	Required
PolicyName	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 658](#) details the pickmap fields for the coverage object.

Table 658. Pick Map Fields for the Coverage Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

Table 659 provides a list of the user key combinations for the coverage object.

Table 659. User Key Fields on the Coverage Object

User Key Field Combinations
CoverageName
IntegrationId
ExternalSystemId

Table 660 details the picklists available for the coverage object.

Table 660. 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 661 details the methods called by the Damage service.

Table 661. Methods Called by Damage Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	DamageDelete
<a href="#">"Execute" on page 116</a>	DamageExecute
<a href="#">"Insert" on page 120</a>	DamageInsert
<a href="#">"QueryPage" on page 123</a>	DamageQueryPage
<a href="#">"Update" on page 145</a>	DamageUpdate

## Fields

[Table 662](#) details the required and read-only fields for the damage object.

Table 662. Required and 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
<a href="#">Audit Fields</a>	Read-only

[Table 663](#) details the pickmap fields for the damage object.

Table 663. Pick Map Fields for the Damage Object

Pick Map Field	Maps To
ClaimExternalId	ClaimId
ClaimIntegrationId	ClaimId

[Table 664](#) provides a list of the user key combinations for the damage object.

Table 664. 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](#), [Book](#), [Financial Account](#), [Financial Account Holder](#), [Financial Account Holding](#), [Financial Plan](#), [Financial Transaction](#), [Lead](#), [Policy](#), [Service Request](#), [Team](#),

## Methods Called

[Table 665](#) details the methods called by the Financial Account service.

Table 665. Methods Called by Financial Account Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	FinancialAccountDelete
<a href="#">"Execute" on page 116</a>	FinancialAccountExecute
<a href="#">"Insert" on page 120</a>	FinancialAccountInsert
<a href="#">"QueryPage" on page 123</a>	FinancialAccountQueryPage
<a href="#">"Update" on page 145</a>	FinancialAccountUpdate

## Fields

[Table 666](#) details the required and read-only fields for the financial account object.

Table 666. Required and Read-Only Fields for the Financial Account Object

Field Name	Type
FinancialAccount	Required
FinancialAccountNumber	Required
Type	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 667](#) details the pick map fields for the financial account object.

Table 667. Pick Map Fields for the Financial Account Object

Pick Map Field	Maps To
ParentFinancialAccountIntegrationId	ParentFinancialAccountId
ParentFinancialAccountExternalSystemId	ParentFinancialAccountId

Table 668 provides a list of the user key combinations for the financial account object.

Table 668. User Key Fields on the Financial Account Object

User Key Field Combinations
FinancialAccount
Id
ExternalSystemId

Table 669 details the picklists available for the financial account object.

Table 669. 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 670 details the methods called by the Financial Account Holder service.

Table 670. Methods Called by Financial Account Holder Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	FinancialAccountHolderDelete
<a href="#">"Execute" on page 116</a>	FinancialAccountHolderExecute
<a href="#">"Insert" on page 120</a>	FinancialAccountHolderInsert
<a href="#">"QueryPage" on page 123</a>	FinancialAccountHolderQueryPage
<a href="#">"Update" on page 145</a>	FinancialAccountHolderUpdate

## Fields

[Table 671](#) details the required and read-only fields for the financial account holder object.

Table 671. Required and Read-Only Fields for the Financial Account Holder Object

Field Name	Type
FinancialAccountHolderName	Required
FinancialAccount	Required
Role	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 672](#) details the pick map fields for the financial account holder object.

Table 672. Pick Map Fields for the Financial Account Holder Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

[Table 673](#) provides a list of the user key combinations for the financial account holder object.

Table 673. User Key Fields on the Financial Account Holder Object

User Key Field Combinations
FinancialAccountHolderName
Id
ExternalSystemId

[Table 674](#) details the picklists available for the financial account holder object.

Table 674. 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 675](#) details the methods called by the Financial Account Holding service.

Table 675. Methods Called by Financial Account Holding Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	FinancialAccountHoldingDelete
<a href="#">"Execute" on page 116</a>	FinancialAccountHoldingExecute
<a href="#">"Insert" on page 120</a>	FinancialAccountHoldingInsert
<a href="#">"QueryPage" on page 123</a>	FinancialAccountHoldingQueryPage
<a href="#">"Update" on page 145</a>	FinancialAccountHoldingUpdate

### Fields

[Table 676](#) details the required and read-only fields for the financial account holding object.

Table 676. Required and Read-Only Fields for the Financial Account Holding Object

Field Name	Type
FinancialAccountHoldingsName	Required
FinancialAccount	Required
FinancialProductFinancialProductName	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only



Table 677 details the pick map fields for the financial account holding object.

Table 677. Pick Map Fields for the Financial Account Holding Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

Table 678 provides a list of the user key combinations for the financial account holding object.

Table 678. 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](#), [Book](#), [Lead](#), [Service Request](#), [Team](#).

### Methods Called

Table 679 details the methods called by the Financial Plan service.

Table 679. Methods Called by Financial Plan Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	FinancialPlanDelete
<a href="#">"Execute" on page 116</a>	FinancialPlanExecute
<a href="#">"Insert" on page 120</a>	FinancialPlanInsert
<a href="#">"QueryPage" on page 123</a>	FinancialPlanQueryPage
<a href="#">"Update" on page 145</a>	FinancialPlanUpdate

## Fields

[Table 680](#) details the required and read-only fields for the financial plan object.

Table 680. Required and Read-Only Fields for the Financial Plan Object

Field Name	Type
FinancialPlan	Required
Status	Required
Type	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 681](#) details the pick map fields for the financial plan object.

Table 681. Pick Map Fields for the Financial Plan Object

Pick Map Field	Maps To
FinancialAccounIntegrationId	FinancialAccountId
FinancialAccounExternalSystemId	FinancialAccountId

[Table 682](#) provides a list of the user key combinations for the financial plan object.

Table 682. User Key Fields on the Financial Plan Object

User Key Field Combinations
FinancialPlan
Id
ExternalSystemId

[Table 683](#) details the picklists available for the financial plan object.

Table 683. 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 684](#) details the methods called by the Financial Product service.

Table 684. Methods Called by Financial Product Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	FinancialProductDelete
<a href="#">"Execute" on page 116</a>	FinancialProductExecute
<a href="#">"Insert" on page 120</a>	FinancialProductInsert
<a href="#">"QueryPage" on page 123</a>	FinancialProductQueryPage
<a href="#">"Update" on page 145</a>	FinancialProductUpdate

### Fields

[Table 685](#) details the required and read-only fields for the financial product object.

Table 685. Required and Read-Only Fields for the Financial Product Object

Field Name	Type
FinancialProductName	Required
CurrencyCode	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 686](#) details the pick map fields for the financial product object.

Table 686. Pick Map Fields for the Financial Product Object

Pick Map Field	Maps To
ParentFinancialProductIntegrationId	ParentFinancialProductId
ParentFinancialProductExternalSystemId	ParentFinancialProductId

Table 687 provides a list of the user key combinations for the financial product object.

Table 687. User Key Fields on the Financial Product Object

User Key Field Combinations
FinancialProductName
Id
ExternalSystemId

Table 688 details the picklists available for the financial product object.

Table 688. 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 689 details the methods called by the Financial Transaction service.

Table 689. Methods Called by Financial Transaction Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	FinancialTransactionDelete
<a href="#">"Execute" on page 116</a>	FinancialTransactionExecute
<a href="#">"Insert" on page 120</a>	FinancialTransactionInsert

Table 689. Methods Called by Financial Transaction Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	FinancialTransactionQueryPage
<a href="#">"Update" on page 145</a>	FinancialTransactionUpdate

## Fields

[Table 690](#) details the required and read-only fields for the financial transaction object.

Table 690. Required and Read-Only Fields for the Financial Transaction Object

Field Name	Type
FinancialAccount	Required
FinancialProductFinancialProductName	Required
TransactionType	Required
TransactionID	Required
<a href="#">Audit Fields</a>	Read-only

[Table 691](#) details the pick map fields for the financial transaction object.

Table 691. Pick Map Fields for the Financial Transaction Object

Pick Map Field	Maps To
FinancialTransactionParentIntegrationId	FinancialTransactionParentId
FinancialTransactionParentExternalSystemId	FinancialTransactionParentId

[Table 692](#) provides a list of the user key combinations for the financial transaction object.

Table 692. User Key Fields on the Financial Transaction Object

User Key Field Combinations
TransactionID
Id
ExternalSystemId

[Table 693](#) details the picklists available for the financial transaction object.

Table 693. 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

[Activity](#), [Contact](#), [Claim](#), [Financial Account](#), [Financial Account Holder](#), [Financial Plan](#), [Lead](#), [Note](#), [Opportunity](#), [Policy](#), [Policy Holder](#), [Portfolio](#), [Service Request](#), [Team](#).

### Methods Called

[Table 694](#) details the methods called by the Household service.

Table 694. Methods Called by Household Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	HouseholdDelete
<a href="#">"Execute" on page 116</a>	HouseholdExecute
<a href="#">"Insert" on page 120</a>	HouseholdInsert
<a href="#">"QueryPage" on page 123</a>	HouseholdQueryPage
<a href="#">"Update" on page 145</a>	HouseholdUpdate

### Fields

[Table 695](#) details the required and read-only fields for the household object.

Table 695. Required and Read-Only Fields for the Household Object

Field Name	Type
HouseholdName	Required
IntegrationID	Required
ExternalSystemID	Required
HouseholdId	Read-only

Table 695. 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
<a href="#">Audit Fields</a>	Read-only

[Table 696](#) details the pick map fields for the household object.

Table 696. Pick Map Field for the Household Object

Pick Map Field	Maps To
ContactExternalId	ContactId
ContactIntegrationId	

Table 697 provides a list of the user key combinations for the household object.

Table 697. User Key Fields on the Household Object

User Key Field Combinations
HouseholdName
Id
ExternalSystemId

Table 698 details the picklists available for the household object.

Table 698. 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 699 details the methods called by the Insurance Property service.

Table 699. Methods Called by Insurance Property Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	InsurancePropertyDelete
<a href="#">"Execute" on page 116</a>	InsurancePropertyExecute
<a href="#">"Insert" on page 120</a>	InsurancePropertyInsert
<a href="#">"QueryPage" on page 123</a>	InsurancePropertyQueryPage
<a href="#">"Update" on page 145</a>	InsurancePropertyUpdate



## Fields

[Table 700](#) details the required and read-only fields for the insurance property object.

Table 700. Required and Read-Only Fields on the Insurance Property Object

Field Name	Type
PolicyName	Required
Type	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 701](#) details the pickmap fields for the insurance property object.

Table 701. Pick Map Fields for the Insurance Property Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

[Table 702](#) provides a list of the user key combinations for the insurance property object.

Table 702. User Key Fields on the Insurance Property Object

User Key Field Combinations
Type
Id
ExternalSystemId

[Table 703](#) details the picklists available for the insurance property object.

Table 703. 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 704](#) details the methods called by the Involved Party service.

Table 704. Methods Called by Involved Party Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	InvolvedPartyDelete
<a href="#">"Execute" on page 116</a>	InvolvedPartyExecute
<a href="#">"Insert" on page 120</a>	InvolvedPartyInsert
<a href="#">"QueryPage" on page 123</a>	InvolvedPartyQueryPage
<a href="#">"Update" on page 145</a>	InvolvedPartyUpdate

### Fields

[Table 705](#) details the required and read-only fields for the involved party object.

Table 705. Required and 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
<a href="#">Audit Fields</a>	Read-only

Table 706 details the pickmap fields for the involved party object.

Table 706. Pick Map Fields for the Involved Party Object

Pick Map Field	Maps To
ClaimExternalSystemId	ClaimId
ClaimIntegrationId	ClaimId

Table 707 provides a list of the user key combinations for the involved party object.

Table 707. User Key Fields on the Involved Party Object

User Key Field Combinations
InvolvedPartyName
Id
ExternalSystemId

Table 708 details the picklists available for the involved party object.

Table 708. 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](#), [Book](#), [Claim](#), [Coverage](#), [Insurance Property](#), [Lead](#), [Policy](#), [Policy Holder](#), [Service Request](#).

## Methods Called

Table 709 details the methods called by the Policy service.

Table 709. Methods Called by Policy Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	PolicyDelete
<a href="#">"Execute" on page 116</a>	PolicyExecute
<a href="#">"Insert" on page 120</a>	PolicyInsert
<a href="#">"QueryPage" on page 123</a>	PolicyQueryPage
<a href="#">"Update" on page 145</a>	PolicyUpdate

## Fields

Table 710 details the required and read-only fields for the policy object.

Table 710. Required and Read-Only Fields on the Policy Object

Field Name	Type
PolicyNumber	Required
PolicyType	Required
CurrencyCode	Read-only
Owner	Read-only
OwnerId	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 711 details the pickmap fields for the policy object.

Table 711. Pick Map Fields for the Policy Object

Pick Map Field	Maps To
ClaimExternalSystemId	ClaimId
ClaimIntegrationId	ClaimId

Table 712 provides a list of the user key combinations for the policy object.

Table 712. User Key Fields on the Policy Object

User Key Field Combinations
PolicyNumber
Id
ExternalSystemId

Table 713 details the picklists available for the policy object.

Table 713. 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 714](#) details the methods called by the Policy Holder service.

Table 714. Methods Called by Policy Holder Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	PolicyHolderDelete
<a href="#">"Execute" on page 116</a>	PolicyHolderExecute
<a href="#">"Insert" on page 120</a>	PolicyHolderInsert
<a href="#">"QueryPage" on page 123</a>	PolicyHolderQueryPage
<a href="#">"Update" on page 145</a>	PolicyHolderUpdate

## Fields

[Table 715](#) details the required and read-only fields for the policy holder object.

Table 715. Required and 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
<a href="#">Audit Fields</a>	Read-only

[Table 716](#) details the pickmap fields for the policy holder object.

Table 716. Pick Map Fields for the Policy Holder Object

Pick Map Field	Maps To
PolicyExternalSystemId	PolicyId
PolicyIntegrationId	PolicyId

Table 717 provides a list of the user key combinations for the policy holder object.

Table 717. User Key Fields on the Policy Holder Object

User Key Field Combinations
PolicyHolderName
Id
ExternalSystemId

Table 718 details the picklists available for the policy holder object.

Table 718. 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.

### Parent Objects

[Contact](#), [CustomObject](#), [Household](#)

### Child Objects

[Contact](#), [Financial Account](#), [Financial Plan](#), [Team](#).

### Methods Called

Table 719 details the methods called by the Portfolio service.

Table 719. Methods Called by Portfolio Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	PortfolioDelete
<a href="#">"Execute" on page 116</a>	PortfolioExecute
<a href="#">"Insert" on page 120</a>	PortfolioInsert

Table 719. Methods Called by Portfolio Service

Method	Name as Defined in Service
<a href="#">"QueryPage" on page 123</a>	PortfolioQueryPage
<a href="#">"Update" on page 145</a>	PortfolioUpdate

## Fields

[Table 720](#) details the read-only fields for the portfolio object.

Table 720. Required and Read-Only Fields on the Portfolio Object

Field Name	Type
PortfolioId	Read-only
Owner	Read-only
OwnerId	Read-only
PrimaryContact	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 721](#) details the pickmap fields for the portfolio object.

Table 721. 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 722](#) provides a list of the user key combinations for the portfolio object.

Table 722. User Key Fields on the Portfolio Object

User Key Field Combinations
AccountNumber and ProductId



Table 722. User Key Fields on the Portfolio Object

User Key Field Combinations
Id
ExternalSystemId

Table 723 details the picklists available for the portfolio object.

Table 723. 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 697](#)
- [“Vehicle” on page 699](#)

### 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 724 details the methods called by the Dealer service.

Table 724. Methods Called by Dealer Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	DealerDelete
<a href="#">"Execute" on page 116</a>	DealerExecute
<a href="#">"Insert" on page 120</a>	DealerInsert
<a href="#">"QueryPage" on page 123</a>	DealerQueryPage
<a href="#">"Update" on page 145</a>	DealerUpdate

## Fields

Table 725 details the required and read-only fields for the dealer object.

Table 725. Required and Read-Only Fields for the Dealer Object

Field Name	Type
DealerId	Required
DealerIntegrationId	Required
DealerExternalSystemID	Required
DealerId	Read-only
DealerType	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 726 details the pick map fields for the dealer object.

Table 726. Pick Map Fields for the Dealer Object

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

Table 727 provides a list of the user key combinations for the dealer object.

Table 727. User Key Fields on the Dealer Object

User Key Field Combinations
DealerId
ExternalSystemID

Table 728 details the picklists available for the dealer object.

Table 728. 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

[Book](#)

### Methods Called

Table 729 details the methods called by the Vehicle service.

Table 729. Methods Called by Vehicle Service

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	VehicleDelete
<a href="#">"Execute" on page 116</a>	VehicleExecute
<a href="#">"Insert" on page 120</a>	VehicleInsert
<a href="#">"QueryPage" on page 123</a>	VehicleQueryPage
<a href="#">"Update" on page 145</a>	VehicleUpdate

## Fields

[Table 730](#) details the required and read-only fields for the vehicle object.

Table 730. 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
<a href="#">Audit Fields</a>	Read-only

[Table 731](#) details the pick map fields for the vehicle object.

Table 731. 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 732](#) provides a list of the user key combinations for the vehicle object.

Table 732. User Key Fields on the Vehicle Object

User Key Field Combinations
Id
ExternalSystemId
ProductId
SerialNumber
ProductExternalId

Table 733 details the picklists available for the vehicle object.

Table 733. 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 v2.0.

The list includes:

- Objects that are child objects only and are not themselves parent objects.
- Objects that are also parent objects.
- The Custom Web Applet child container.

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 734](#). The methods that you can use on a child object are the same as for its parent object.

Table 734. Methods Called On Child Objects

Method	Name as Defined in Service
<a href="#">"Delete" on page 112</a>	<object>Delete
<a href="#">"Execute" on page 116</a>	<object>Execute
<a href="#">"Insert" on page 120</a>	<object>Insert
<a href="#">"QueryPage" on page 123</a>	<object>QueryPage
<a href="#">"Update" on page 145</a>	<object>Update

**NOTE:** For the Custom Web Applet child, only the QueryPage method is supported.

The following child objects are listed:

- ["Account" on page 703](#)
- ["Account Contact" on page 704](#)
- ["Account Opportunity" on page 706](#)
- ["Activity" on page 707](#)
- ["Address" on page 709](#)
- ["Asset" on page 712](#)
- ["Attachment" on page 714](#)
- ["Attendee Call" on page 715](#)
- ["Book" on page 717](#)
- ["Book User" on page 718](#)
- ["Campaign" on page 720](#)
- ["Competitor" on page 721](#)
- ["Contact" on page 722](#)
- ["Contact Role" on page 724](#)
- ["Custom Object 1 through 3" on page 725](#)
- ["Custom Object 4 through 15" on page 726](#)
- ["Custom Object 1 Child" on page 727](#)
- ["Custom Web Applet" on page 728](#)
- ["Delegated User" on page 729](#)
- ["Exchange Rate History" on page 729](#)
- ["Image" on page 730](#)

- ["Interests" on page 731](#)
- ["Invitee" on page 732](#)
- ["Lead" on page 734](#)
- ["Note" on page 735](#)
- ["Opportunity" on page 736](#)
- ["Opportunity Child" on page 737](#)
- ["Products Detailed" on page 737](#)
- ["Promotional Items Dropped" on page 739](#)
- ["Recipient" on page 740](#)
- ["Related Contact" on page 741](#)
- ["Revenue" on page 742](#)
- ["Sales Assessment" on page 744](#)
- ["Sales Assessment Value" on page 745](#)
- ["Sample Dropped" on page 745](#)
- ["Service Request" on page 747](#)
- ["Shared Address" on page 748](#)
- ["Subbook" on page 752](#)
- ["Team" on page 753](#)
- ["User" on page 754](#)
- ["User Default Book Per Type" on page 755](#)

**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

[Address](#), [Contact](#), [CustomObject](#)

## Fields

[Table 735](#) details the required and read-only fields for the account child object.

Table 735. Required and Read-Only Fields for the Account Child Object

Field Name	Type
AccountId	Read-only

[Table 736](#) details the pick map fields for the account child object.

Table 736. Pick Map Fields for the Account Child Object

Pick Map Field	Maps To
ExternalSystemId	AccountId

[Table 737](#) provides a list of the user key combinations for the account child object.

Table 737. User Key Fields on the Account Child Object

User Key Field Combinations
AccountId
ExternalSystemId
Name and Location

[Table 738](#) details the picklists available for the account child object.

Table 738. 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 739](#) details the required and read-only fields for the account contact child object.

Table 739. Required and Read-Only Fields for the Account Contact Child Object

Field Name	Type
AccountLocation	Read-only
AccountMainPhone	Read-only
AccountType	Read-only

[Table 740](#) details the pick map fields for the account contact child object.

Table 740. 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 741](#) provides a list of the user key combinations for the account contact child object.

Table 741. User Key Fields on the Account Contact Child Object

User Key Field Combinations
AccountId
AccountExternalSystemId
AccountLocation and AccountName
ContactExternalSystemId
ContactFirstName and ContactLastName

Table 742 details the picklists available for the account contact child object.

Table 742. 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 743 details the required and read-only fields for the account opportunity child object.

Table 743. 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 744 details the pick map fields for the account opportunity child object.

Table 744. 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 745 provides a list of the user key combinations for the account opportunity child object.

Table 745. User Key Fields on the Account Opportunity Child Object

User Key Field Combinations
OpportunityId
Id

## Activity

The activity object stores information on an activity that a user must carry out, for example, a callback activity for an account.

For activity child objects of Custom Object 01 through Custom Object 03, you can use the Activity field to query for tasks and appointments. The field can have the value Task or Appointment.

## Parent Objects

[Account](#), [Campaign](#), [Contact](#), [CustomObject](#), [Household](#), [Lead](#), [Opportunity](#), [Service Request](#).

## Fields

Table 746 details the required and read-only fields for the activity child object.

Table 746. Required and Read-Only Fields for the Activity Child Object

Field Name	Type
AccountLocation	Read-only
CreatedDetail	Read-only
MEEventName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
CODInteractionTime	Read-only
CODWrapUpTime	Read-only
CODHandleTime	Read-only
CODIVRTIME	Read-only
CODQueueHoldTime	Read-only
CODTotalHoldTime	Read-only
DescriptionShadow	Read-only
DealerName	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 747 details the pick map fields for the activity child object.

Table 747. 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

Table 747. Pick Map Fields for the Activity Child Object

Pick Map Field	Maps To
PrimaryContactExternalId	PrimaryContactId
LeadIntegrationId	LeadId
LeadExternalId	LeadId
OwnerExternalId	OwnerId
OwnerIntegrationId	OwnerId
SRIntegrationId	SRId
SRExternalId	SRId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 748 provides a list of the user key combinations for the activity child object.

Table 748. 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 748](#). To create and update shared addresses, the top-level Address Web service must be used. For more information, see ["Address" on page 527](#).

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 749 on page 711](#), 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 519](#), and [“Adding and Updating Contact Addresses” on page 548](#).

For information about converting unshared addresses to shared addresses using UI export functionality and Web services calls, see [“Converting Unshared Addresses to Shared Addresses” on page 528](#).

Table 749. 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 <a href="#">"Address" on page 527</a> .	Cannot be used because addresses are being shared. To create and update addresses, the top-level Address Web service must be used. see <a href="#">"Address" on page 527</a> .	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 <a href="#">"Adding and Updating Account Addresses" on page 519</a> .

## Parent Objects

[Account](#), [Contact](#)

## Fields

[Table 750](#) details the required and read-only fields for the address child object.

Table 750. Required and Read-Only Fields for the Address Child Object

Field Name	Type
AddressId	Read-only
GeocodeMatchCode	Read-only
GeocodeMatchVector	Read-only
GeocodeStatus	Read-only

[Table 751](#) details the pick map fields for the address child object.

Table 751. Pick Map Fields for the Address Child Object

Pick Map Field	Maps To
None	

[Table 752](#) provides a list of the user key combinations for the address child object.

Table 752. 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](#), [CustomObject](#)



## Fields

[Table 753](#) details the required and read-only fields for the asset child object.

Table 753. 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 754](#) details the pick map fields for the asset child object.

Table 754. Pick Map Fields for the Asset Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
Manufacturer	ManufacturerId
PreferredServiceDealer	PreferredServiceDealerID

[Table 755](#) provides a list of the user key combinations for the asset child object.

Table 755. 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](#), [Event](#), [Exam](#), [Lead](#), [Opportunity](#), [Service Request](#), and [Solution](#).

### Fields

[Table 756](#) details the required and read-only fields for the attachment object.

Table 756. Required and Read-Only Fields for the Attachment Child Object

Field Name	Type
DisplayFileName	Required
FileNameOrURL	Required
FileDate	Read-only
FileSize	Read-only
AttachmentTag	Read-only (accessible for the Attachment child of Account and Contact only)
<parent_object>Id	Read-only
Id	Read-only
ModId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 757](#) provides a list of the user key combinations for the attachment child object.

Table 757. 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 758](#) details the required and read-only fields for the attendee call child object.

Table 758. Required and Read-Only Fields for the Attendee Call Child Object

Field Name	Type
Subject	Required
PrimaryContact	Required
<a href="#">Audit Fields</a>	Read-only

[Table 759](#) details the pick map fields for the attendee call child object.

Table 759. 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

Table 759. Pick Map Fields for the Attendee Call Child Object

Pick Map Field	Maps To
DelegatedBy	DelegatedById
DelegatedByExternalSystemId	DelegatedById
DelegatedByExternalSystemId	DelegatedByFirstName
DelegatedByExternalSystemId	DelegatedByLastName
DealRegistrationName	DealRegistrationId
DealRegistrationName	DealRegistrationExternalSystemId
FinancialAccountExternalSystemId	FinancialAccountFinancialAccount
FinancialPlanExternalSystemId	FinancialPlanFinancialPlan
FundRequest	FundRequestId
Lead	LeadId
LeadIntegrationId	LeadId
LeadExternalSystemId	LeadId
LeadExternalSystemId	LeadFirstName
LeadExternalSystemId	LeadLastName
MDFRequestExternalSystemId	MDFRequestName
MEEventName	MEEventId
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

Table 759. Pick Map Fields for the Attendee Call Child Object

Pick Map Field	Maps To
ServiceRequestExternalSystemId	ServiceRequestId
ServiceRequestExternalSystemId	ServiceRequestNumber
SPRequestExternalSystemId	SPRequestName

Table 760 provides a list of the user key combinations for the attendee call child object.

Table 760. User Key Fields on the Attendee Call Child Object

User Key Field Combinations
Id
PrimaryContactId
PrimaryContact
PrimaryContactExternalId

Table 761 provides a list of the picklists for the attendee call child object.

Table 761. 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 Objects

Account, Accreditation, Allocation, Application, Business Plan, Campaign, Certification, Contact, Course, CustomObject, Deal Registration, Event, Exam, Financial Account, Financial Plan, HCP Contact Allocation, Inventory Audit Report, Inventory Period, Lead, MDF Request, Messaging Plan, Objective, Opportunity, Order, Partner, Policy, Solution, Sample Lot, Sample Request, Sample Transaction, Special Pricing Request, Vehicle.

## Fields

Table 762 details the required and read-only fields for the book child object.

Table 762. Required and Read-Only Fields for the Book Child Object

Field Name	Type
BookName	Required
BookId	Read-only
SystemAssociateFlag	Read-only
ModId	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 763 details the pick map fields for the book child object.

Table 763. Pick Map Fields for the Book Child Object

Pick Map Field	Maps To
BookName	BookId

Table 764 provides a list of the user key combinations for the book child object.

Table 764. User Key Fields on the Book Child Object

User Key Field Combinations
BookId
BookName

## Book User

The book user object stores information about a book user.

## Parent Object

Book

## Fields

Table 765 details the required and read-only fields for the book user child object.

Table 765. Required and Read-Only Fields for the Book User Child Object

Field Name	Type
AccessProfileName	Required
UserAlias	Required
UserFirstName	Read-only
UserLastName	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 766 details the pick map fields for the book user child object.

Table 766. Pick Map Fields for the Book User Child Object

Pick Map Field	Maps To
BookExternalSystemId	BookId
UserAlias	UserId
UserEMailAddr	UserId
UserExternalSystemId	UserId
UserSignInId	UserId

Table 767 provides a list of the user key combinations for the book user child object.

Table 767. User Key Fields on the Book User Child Object

User Key Field Combinations
Id
UserId
UserAlias
UserExternalSystemId

Table 768 provides a list of the picklists for the book user child object.

Table 768. Picklists for the Book User Child Object

Field Name
BookRole

## Campaign

The campaign object stores information on the recipients of campaigns.

### Parent Object

[Contact](#), [Event](#)

### Fields

Table 769 details the required and read-only fields for the campaign child object.

Table 769. Required and Read-Only Fields for the Campaign Child Object

Field Name	Type
CampaignName	Read-only
CampaignContactId	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 770 details the pick map fields for the campaign child object.

Table 770. Pick Map Fields for the Campaign Child Object

Pick Map Field	Maps To
CampaignExternalSystemId	CampaignId
CampaignName	CampaignId



Table 771 provides a list of the user key combinations for the campaign child object.

Table 771. User Key Fields on the Campaign Child Object

User Key Field Combinations
CampaignId
CampaignExternalId
CampaignName

Table 772 details the picklists available for the campaign child object.

Table 772. 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 Objects

[Account](#), [Opportunity](#)

## Fields

Table 773 details the required and read-only fields for the competitor child object.

Table 773. Required and Read-Only Fields for the Competitor Child Object

Field Name	Type	Parent Object
CompetitorId	Required	Both
StartDate	Required	Both

Table 774 details the pick map fields for the competitor child object.

Table 774. 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 775 provides a list of the user key combinations for the competitor child object.

Table 775. 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 776 details the picklists available for the competitor child object.

Table 776. 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 Objects

[Activity](#), [Address](#), [Account](#), [Campaign](#), [CustomObject](#), [Event](#), [Household](#), [Opportunity](#), [Portfolio](#)

## Fields

[Table 777](#) details the required and read-only fields for the contact child object.

Table 777. Required and Read-Only Fields for the Contact Child Object

Field Name	Type
ContactId	Read-only
ContactFirstName	Read-only
ContactLastName	Read-only
ContactAccountId	Read-only
ContactAccountName	Read-only
ContactAccountLocation	Read-only
ContactAccountIntegrationId	Read-only
ContactAccountExternalSystemId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 778](#) details the pick map fields for the contact child object.

Table 778. Pick Map Fields for the Contact Child Object

Pick Map Field	Maps To
ContactExternalId	ContactId
ContactIntegrationId	ContactId

[Table 779](#) provides a list of the user key combinations for the contact child object.

Table 779. 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 Objects

[Account](#), [Opportunity](#)

## Fields

[Table 780](#) details the required and read-only fields for the contact role child object.

Table 780. Required and Read-Only Fields for the Contact Role Child Object

Field Name	Type	Parent Object
AccountId	Required	Both
ContactId	Required	Opportunity

[Table 781](#) details the pick map fields for the contact role child object.

Table 781. 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 782 provides a list of the user key combinations for the contact role child object.

Table 782. 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 783 details the picklists available for the contact role child object.

Table 783. Picklists Available for the Contact Role Child Object

Field Name	Parent Object
Role	Account
BuyingRole	Opportunity

## Custom Object 1 through 3

The custom objects 1, 2, and 3.

### Parent Objects

[Account](#), [Activity](#), [Address](#), [Campaign](#), [Contact](#), [Lead](#), [Message Response](#), [Messaging Plan](#), [Messaging Plan Item](#), [Messaging Plan Item Relation](#), [Opportunity](#), [Service Request](#).

### Fields

Table 784 details the required and read-only fields for the custom object child object.

Table 784. Required and Read-Only Fields for the Custom Object Child Object

Field Name	Type
Name	Required
Id	Read-only

Table 785 provides a list of the user key combinations for the custom object child object.

Table 785. 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 786 details the required and read-only fields for the custom object child object.

Table 786. Required and Read-Only Fields for the Custom Object Child Object

Field Name	Type
Name	Required
Currency	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 787 details the pick map fields for the custom object child object.

Table 787. Pick Map Fields for the Custom Object Child Object

Pick Map Field	Maps To
Owner	OwnerId

Table 788 provides a list of the user key combinations for the custom object child object.

Table 788. User Key Fields on the Custom Object Child Object

User Key Field Combinations
Id
ExternalSystemId

Table 789 details the picklists available for the custom object child object.

Table 789. Picklists Available for the Custom Object Child Object

Field Name
Type

## Custom Object 1 Child

The Custom Object 1 child object.

### Parent Objects

Custom Object 1

### Fields

Table 790 details the required and read-only fields for the Custom Object 1 child object.

Table 790. Required and Read-Only Fields for the Custom Object 1 Child Object

Field Name	Type
Name	Required
Id	Read-only

Table 791 provides a list of the user key combinations for the Custom Object 1 child object.

Table 791. User Key Fields on the Custom Object 1 Child Object

User Key Field Combinations
Id

Table 791. User Key Fields on the Custom Object 1 Child Object

User Key Field Combinations
ExternalSystemId
Name

No pickmap or picklist fields are available.

## Custom Web Applet

For parent objects that support custom Web applets, you can use the QueryPage method to access the Detail page custom Web applets for the object. All custom Web applets enabled for the company can be returned regardless of whether they are enabled in the page layout for the record type.

The URL and HTML code in custom Web applets can contain field and function parameters that are evaluated at run time. The QueryPage method returns URL and HTML data with the evaluated fields and functions. This contrasts with the Administrative Web service methods, DetailPageCustomWebAppletRead and DetailPageCustomWebAppletReadAll, which do not return evaluated URLs or HTML data.

Table 792 shows the custom Web applet fields that are returned by the QueryPage method and whether they are filterable:

Table 792. Custom Web Applet Fields

Field Name	Description	Filterable
Name	The language-independent name of the custom Web applet.	Yes
DisplayName	The display name of the Web applet in the language of the currently logged-in user.	Yes
Type	The type of Web applet: Feed, HTML, or URL.	Yes
URL	The evaluated URL for the Feed or URL type of Web applet.	No
HTMLHead	The evaluated HTML header code for the HTML type of Web applet.	No
HTMLBody	The evaluated HTML body code for the HTML type of Web applet.	No
Height	The height of the Web applet in pixels.	Yes

Other fields, similar to the audit fields returned for record type objects are also returned for each custom Web applet. For more information, see [“Audit Fields” on page 375](#).

For more information about custom Web applets, see *Oracle CRM On Demand Online Help*.



## 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 793 details the required and read-only fields for the delegated user child object.

Table 793. Required and Read-Only Fields for the Delegated User Child Object

Field Name	Type
UserId	Required
UserRole	Required
<a href="#">Audit Fields</a>	Read-only

Table 794 provides a list of the user key combinations for the delegated user child object.

Table 794. User Key Fields on the Delegated User Child Object

User Key Field Combinations
Id
ExternalSystemId

Table 795 details the picklists available for the delegated user child object.

Table 795. 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 796](#) details the required and read-only fields for the exchange rate object.

Table 796. Required and Read-Only Fields for the Exchange Rate History Object

Field Name	Type
FromCurrencyCode	Required
ToCurrencyCode	Required
ExchangeDate	Required
ExchangeRate	Required

[Table 797](#) provides a list of the user key combinations for the exchange rate history object.

Table 797. User Key Fields on the Exchange Rate History Object

User Key Field Combinations
FromCurrencyCode
ToCurrencyCode
ExchangeDate

## Image

The image object stores an image that is displayed on detail pages for the Account, Contact, and Product record types. You can insert, update, query, and delete images using Web service calls.

An image can only be inserted when the Display Image check box on the Customize Record Types page is selected for the record type. An image cannot be inserted into an account that already has an associated image.

The maximum file size for an image is 5 MB, and the maximum dimensions are 150 pixels high and 150 pixels wide; larger images will be cropped to 150 x 150 pixels.

Supported image formats are: JPEG, JPG, BMP, PNG, and GIF. If you specify any other format, or a blank value for the <FileExtension> element, an error is returned.

Deleted images are not moved to the Deleted Items area, so they cannot be restored.

## Parent Object

[Account](#), [Contact](#), [Product](#)

## Fields

[Table 798](#) details the required and read-only fields for the image child object.

Table 798. Required and Read-Only Fields for the Image Child Object

Field Name	Type
Attachment	Required
DisplayFileName	Required
FileNameOrURL	Required
<a href="#">Audit Fields</a>	Read-only

[Table 799](#) provides a list of the user key combinations for the image child object.

Table 799. User Key Fields on the Image Child Object

User Key Field Combinations
Id
ExternalSystemId
FileNameOrURL and FileExtension

## 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 800](#) details the required and read-only fields for the interests child object.

Table 800. Required and Read-Only Fields for the Interests Child Object

Field Name	Type
Category	Required
Interests	Required
InterestId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 801](#) details the pick map fields for the interests child object.

Table 801. Pick Map Fields for the Interests Child Object

Pick Map Field	Maps To
InterestExternalSystemId	InterestId

[Table 802](#) provides a list of the user key combinations for the interests child object.

Table 802. User Key Fields on the Interests Child Object

User Key Field Combinations
InterestId
ExternalSystemId

[Table 803](#) details the picklists available for the interests child object.

Table 803. Picklists Available for the Interests Child Object

Field Name
Category
Subjects

## Invitee

The invitee object stores information about invitees to events, including feedback about the invitation.

## Parent Object

Event

## Fields

Table 804 details the required and read-only fields for the invitee child object.

Table 804. Required and Read-Only Fields for the Invitee Child Object

Field Name	Type
ContactFullName	Required
InviteeStatus	Required
InviteeId	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 805 details the pick map fields for the invitee child object.

Table 805. Pick Map Fields for the Invitee Child Object

Pick Map Field	Maps To
InviteeExternalSystemId	InviteeId

Table 806 provides a list of the user key combinations for the invitee child object.

Table 806. User Key Fields on the Invitee Child Object

User Key Field Combinations
ContactIdExternalId
InviteeId

Table 807 details the picklists available for the invitee child object.

Table 807. 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](#), [Household](#), [Opportunity](#)

## Fields

[Table 808](#) details the required and read-only fields for the lead child object.

Table 808. Required and Read-Only Fields for the Lead Child Object

Field Name	Type
OwnerFullName	Read-only
SalesRepFullName	Read-only
CreatedbyEmailAddress	Read-only
ModifiedbyEmailAddress	Read-only
ReferredByFullName	Read-only
FuriganaAccountName	Read-only
FuriganaContactFirstName	Read-only
FuriganaContactLastName	Read-only
FirstNameShadow	Read-only
LastNameShadow	Read-only
CompanyNameShadow	Read-only
AssignmentStatus	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only

[Table 809](#) details the pick map fields for the lead child object.

Table 809. Pick Map Fields for the Lead Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId

Table 809. Pick Map Fields for the Lead Child Object

Pick Map Field	Maps To
AccountLocation	AccountId
AccountName	AccountId
CampaignExternalId	CampaignId
CampaignIntegrationId	CampaignId
CampaignName	CampaignId
ContactExternalId	ContactId
ContactIntegrationId	ContactId
OpportunityIntegrationId	OpportunityId
OpportunityExternalId	OpportunityId
OwnerExternalId	OwnerId
OwnerIntegrationId	OwnerId
SalesRepAlias	SalesRepId
CustomObject1Name	CustomObject1Id
CustomObject2Name	CustomObject2Id
CustomObject3Name	CustomObject3Id

Table 810 provides a list of the user key combinations for the lead child object.

Table 810. 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.

## Usage

Users who have the View Private Notes privilege enabled in their user role can make Web Service calls to read all child-level private notes, regardless of the owner of the note. Private notes are those that have the Private check box selected in the UI. You can read the child-level notes only using a parent object's QueryPage operation (for example, ContactQueryPage). The notes are retrieved through the <ListOfNote> element.

The View Private Notes privilege does not allow a user to update or delete a note that is owned by another user. Typically, the View Private Notes privilege is given to company administrators only.

## Parent Objects

[Account](#), [Campaign](#), [Contact](#), [CustomObject](#), [Household](#), [Opportunity](#), [Service Request](#)

## Fields

[Table 811](#) details the required and read-only fields for the note child object.

Table 811. Required and Read-Only Fields for the Note Child Object

Field Name	Type
Subject	Required
NotelId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 812](#) provides a list of the user key combinations for the note child object.

Table 812. User Key Fields on the Note Child Object

User Key Field Combinations
NotelId

## 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](#), [CustomObject](#), [Household](#), [Partner](#)



## Fields

[Table 813](#) details the required and read-only fields for the opportunity child object.

Table 813. Required and Read-Only Fields for the Opportunity Child Object

Field Name	Type
OpportunityName	Required
CloseDate	Required

[Table 814](#) provides a list of the user key combinations for the opportunity child object.

Table 814. 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](#)

## 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.

## Usage

Product blocking rules are respected for this object. A blocked product is a product for which your company does not want users to provide any product information, samples, or promotional items for that product to the associated contact.

Web service requests cannot insert, update, or delete records for this object if the product specified in the request is a blocked product. An error message is generated when such a request is made. For more information about blocked products, see *Oracle CRM On Demand Online Help* and [“Blocked Product” on page 616](#).

You can access call objective, outcome, and next call objective records associated to the ProductID of a products detailed record through a Web services call. For more information about these structured product messaging record types, see *Oracle CRM On Demand Online Help*.

In the Oracle CRM On Demand UI, if the company administrator has selected the Enable Filtering of Product Selection by Sales Rep Allocation check box, then for products detailed only the products that are allocated to the sales representative appear in the Product selector. However, in Web service calls it is possible to specify a product of any type regardless of whether the check box is selected. Be aware that it is your responsibility to validate data before making Web service calls. For more information about filtering of product selection by sales representative allocation, see *Oracle CRM On Demand Online Help*.

### Parent Object

[Activity](#)

### Fields

[Table 815](#) details the required and read-only fields for the products detailed child object.

Table 815. Required and Read-Only Fields for the Products Detailed Child Object

Field Name	Type
ProductId	Required
Indication	Required
ProductDetailedId	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 816](#) details the pick map fields for the products detailed child object.

Table 816. Pick Map Fields for the Products Detailed Child Object

Pick Map Field	Maps To
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

Table 817 provides a list of the user key combinations for the products detailed child object.

Table 817. User Key Fields on the Products Detailed Child Object

User Key Field Combinations
ProductDetailedId
ProductDetailedExternalSystemId
ProductExternalSystemId
ProductId
ProductExternalSystemId
Name
Indication

## Promotional Items Dropped

The promotional items dropped object stores the information on promotional items dropped for an activity.

### Usage

Product blocking rules are respected for this object. A blocked product is a product for which your company does not want users to provide any product information, samples, or promotional items for that product to the associated contact.

Web service requests cannot insert, update, or delete records for this object if the product specified in the request is a blocked product. An error message is generated when such a request is made. For more information about blocked products, see *Oracle CRM On Demand Online Help* and [“Blocked Product” on page 616](#).

In the Oracle CRM On Demand UI, if the company administrator has selected the Enable Filtering of Product Selection by Sales Rep Allocation check box, then for promotional items dropped only the products that are allocated to the sales representative appear in the Product selector. However, in Web service calls it is possible to specify a product of any type regardless of whether the check box is selected. Be aware that it is your responsibility to validate data before making Web service calls. For more information about filtering of product selection by sales representative allocation, see *Oracle CRM On Demand Online Help*.

### Parent Object

[Activity](#)

## Fields

[Table 818](#) details the required and read-only fields for the promotional items dropped child object.

Table 818. Required and Read-Only Fields for the Promotional Items Dropped Child Object

Field Name	Type
ProductName	Required
Quantity	Required
ProductAllocationId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 819](#) details the pick map fields for the promotional items dropped child object.

Table 819. Pick Map Fields for the Promotional Items Dropped Child Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

[Table 820](#) provides a list of the user key combinations for the promotional items dropped child object.

Table 820. 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 821](#) details the required and read-only fields for the recipient child object.

Table 821. Required and Read-Only Fields for the Recipient Child Object

Field Name	Type
ContactId	Required
ModifiedDate	Read-only

[Table 822](#) provides a list of the user key combinations for the recipient child object.

Table 822. 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 823](#) details the required and read-only fields for the related contact child object.

Table 823. 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 824 details the pick map fields for the related contact child object.

Table 824. Pick Map Fields for the Related Contact Child Object

Pick Map Field	Maps To
RelatedContactExternalId	RelatedContactId
RelatedContactIntegrationId	RelatedContactId

Table 825 provides a list of the user key combinations for the related contact child object.

Table 825. User Key Fields on the Related Contact Child Object

User Key Field Combinations
None

Table 826 details the picklists available for the related contact child object.

Table 826. 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 827](#) details the required and read-only fields for the revenue child object.

Table 827. Required and Read-Only Fields for the Revenue Child Object

Field Name	Type
RevenueId	Required
PartNumber	Required
Revenue	Required
ContactFullName	Required
<a href="#">Audit Fields</a>	Read-only

[Table 828](#) details the pick map fields for the revenue child object.

Table 828. Pick Map Fields for the Revenue Child Object

Pick Map Field	Maps To
Product	ProductId
ProductExternalId	ProductId
ProductIntegrationId	ProductId
ProductCategory	ProductCategoryId
ProductCategoryExternalId	ProductCategoryId
ProductCategoryIntegrationId	ProductCategoryId

[Table 829](#) provides a list of the user key combinations for the revenue child object.

Table 829. User Key Fields on the Revenue Child Object

User Key Field Combinations
RevenueId
ProductId
ProductExternalId

Table 830 details the picklists available for the revenue child object.

Table 830. 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, contact, or objective.

### Parent Objects

[Account](#), [Activity](#), [Business Plan](#), [Contact](#), [Objective](#), [Opportunity](#), [Service Request](#).

### Fields

Table 831 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 831. Required and Read-Only Fields for the Sales Assessment Child Object

Field Name	Type
TemplateId	Required
Name	Required
<a href="#">Audit Fields</a>	Read-only

Table 832 provides a list of the user key combinations for the sales assessment child object.

Table 832. 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 Objects

[Account](#), [Activity](#), [Business Plan](#), [Contact](#), [Objective](#), [Opportunity](#), [Service Request](#).

### Fields

[Table 833](#) details the required and read-only fields for the sales assessment value child object.

Table 833. Required and Read-Only Fields for the Sales Assessment Value Child Object

Field Name	Type
AssessId	Required
AttribId	Required
<a href="#">Audit Fields</a>	Read-only

[Table 834](#) provides a list of the user key combinations for the sales assessment value child object.

Table 834. User Key Fields on the Sales Assessment Value Child Object

User Key Field Combinations
AssessId
AttribId

## Sample Dropped

The sample dropped 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.

### Usage

Product blocking rules are respected for this object. A blocked product is a product for which your company does not want users to provide any product information, samples, or promotional items for that product to the associated contact.

Web service requests cannot insert, update, or delete records for this object if the product specified in the request is a blocked product. An error message is generated when such a request is made. For more information about blocked products, see *Oracle CRM On Demand Online Help* and [“Blocked Product” on page 616](#).

In the Oracle CRM On Demand UI, if the company administrator has selected the Enable Filtering of Product Selection by Sales Rep Allocation check box, then for samples dropped only the products that are allocated to the sales representative appear in the Product selector. However, in Web service calls it is possible to specify a product of any type regardless of whether the check box is selected. Be aware that it is your responsibility to validate data before making Web service calls. For more information about filtering of product selection by sales representative allocation, see *Oracle CRM On Demand Online Help*.

## Parent Object

[Activity](#)

## Fields

[Table 835](#) details the required and read-only fields for the sample dropped child object.

Table 835. Required and Read-Only Fields for the Sample Dropped Child Object

Field Name	Type
ProductId	Required
Quantity	Required
Id	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 836](#) details the pick map fields for the sample dropped child object.

Table 836. Pick Map Fields for the Sample Dropped Child Object

Pick Map Field	Maps To
ProductName	ProductId
ProductExternalSystemId	ProductId
ProductIntegrationId	ProductId

[Table 837](#) provides a list of the user key combinations for the sample dropped child object.

Table 837. User Key Fields on the Sample Dropped Child Object

User Key Field Combinations
Id
SampleDroppedExternalSystemId

## 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](#), [Household](#), [Opportunity](#), [Solution](#)

### Fields

[Table 838](#) details the required and read-only fields for the service request child object.

Table 838. 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
SRNumberShadow	Read-only
AbstractShadow	Read-only
LastAssignmentCompletionDate	Read-only
LastAssignmentSubmissionDate	Read-only
<a href="#">Audit Fields</a>	Read-only

[Table 839](#) details the pick map fields for the service request child object.

Table 839. Pick Map Fields for the Service Request Child Object

Pick Map Field	Maps To
AccountIntegrationId	AccountId
AccountExternalId	AccountId

Table 839. Pick Map Fields for the Service Request Child Object

Pick Map Field	Maps To
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 840 provides a list of the user key combinations for the service request child object.

Table 840. 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 527](#). 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 709](#). See [Table 841 on page 750](#) for information about the usage of address child objects and fields depending on whether addresses are shared. For information about converting unshared addresses to shared addresses using UI export functionality and Web services calls, see ["Converting Unshared Addresses to Shared Addresses" on page 528](#).

Table 841. 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 <a href="#">"Address" on page 527</a> .	Cannot be used because addresses are being shared. To create and update addresses, the top-level Address Web service must be used. see <a href="#">"Address" on page 527</a> .	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 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 and change in address association are not allowed, but query operations are not restricted.	Can be used.	Usage is as described in <a href="#">"Adding and Updating Account Addresses" on page 519</a> .

## Parent Objects

[Account](#), [Contact](#), [Dealer](#), [Partner](#)

## Fields

[Table 842](#) details the required and read-only fields for the shared address child object.

Table 842. Required and Read-Only Fields for the Shared Address Child Object

Field Name	Type
AddressId	Read-only

[Table 843](#) details the pick map fields for the shared address child object.

Table 843. Pick Map Fields for the Shared Address Child Object

Pick Map Field	Maps To
AddressName	AddressId
AddressExternalSystemId	AddressId

[Table 844](#) provides a list of the user key combinations for the shared address child object.

Table 844. User Key Fields on the Shared Address Child Object

User Key Field Combinations
Id
AddressId
AddressName
AddressExternalSystemId

[Table 845](#) details the picklists available for the shared address child object.

Table 845. Picklists Available for the Shared Address Child Object

Field Name
AddressType
CurrencyCode

## Subbook

The subbook object stores information about a subbook.

### Usage

After a partner has been associated with a subbook through the PartnerId field, the association cannot be changed.

### Parent Object

[Book](#)

### Fields

[Table 846](#) details the required and read-only fields for the subbook child object.

Table 846. Required and Read-Only Fields for the Subbook Child Object

Field Name	Type
BookName	Required
<a href="#">Audit Fields</a>	Read-only

[Table 847](#) details the pick map fields for the subbook child object.

Table 847. Pick Map Fields for the Subbook Child Object

Pick Map Field	Maps To
ParentBookName	ParentBookId

[Table 848](#) provides a list of the user key combinations for the subbook child object.

Table 848. User Key Fields on the Subbook Child Object

User Key Field Combinations
Id
BookName



## Team

The team object stores information on the team that is assigned to a particular record type. In this way, a team of employees can, for example, 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](#), [Accreditation](#), [Application](#), [Business Plan](#), [Certification](#), [Contact](#), [Course](#), [CustomObject](#), [Deal Registration](#), [Event](#), [Exam](#), [Financial Account](#), [Financial Plan](#), [Household](#), [Lead](#), [MDF Request](#), [Objective](#), [Opportunity](#), [Order](#), [Partner](#), [Portfolio](#), [Sample Request](#), [Service Request](#), [Special Pricing Request](#).

### Fields

[Table 849](#) details the required and read-only fields for the team child object.

Table 849. Required and Read-Only Fields for the Team Child Object

Field Name	Type
<i>record_typeAccess</i>	Required
UserId	Required
LoginName	Read-only
UserFirstName	Read-only

[Table 850](#) details the pick map fields for the team child object.

Table 850. Pick Map Fields for the Team Child Object

Pick Map Field	Maps To
UserLastName	UserFirstName
<i>record_typeAccess</i>	<i>record_typeAccessId</i>
UserAlias	UserId
UserEmail	UserId
UserExternalSystemId	UserId
FullName	UserId

Table 851 provides a list of the user key combinations for the team child object.

Table 851. User Key Fields on the Team Child Object

User Key Field Combinations
UserId
UserExternalSystemId
UserEmail
UserAlias

Table 852 details the picklists available for the team child object.

Table 852. Picklists Available for the Team Child Object

Field Name
TeamRole

## 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 853 details the required and read-only fields for the user child object.

Table 853. 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
<a href="#">Audit Fields</a>	Read-only

Table 854 details the pick map fields for the user child object.

Table 854. Pick Map Fields for the User Child Object

Pick Map Field	Maps To
UserExternalId	UserId
UserIntegrationId	UserId

Table 855 provides a list of the user key combinations for the user child object.

Table 855. User Key Fields on the User Child Object

User Key Field Combinations
UserId
UserExternalId

## User Default Book Per Type

The user default book per type object stores information about the default book for a user for each record type. The record type is identified by the access object name.

### Parent Object

User

### Fields

Table 856 details the required and read-only fields for the user default book per type child object.

Table 856. Required and Read-Only Fields for the User Default Book Per Type Child Object

Field Name	Type
AccessObjectName	Required
BookId	Either one of these fields is required
BookName	
BookDescription	Read-only
ObjectName	Read-only
<a href="#">Audit Fields</a>	Read-only

Table 857 details the pick map fields for the user default book per type child object.

Table 857. Pick Map Fields for the User Default Book Per Type Child Object

Pick Map Field	Maps To
BookName	BookId
BookExternalSystemId	BookId

Table 858 provides a list of the user key combinations for the user default book per type child object.

Table 858. User Key Fields on the User Default Book Per Type Child Object

User Key Field Combinations
Id

# 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 757](#)
- [About The Attachment Element on page 757](#)
- [Specifying File Attachment Content on page 759](#)
- [Retrieving or Querying File Attachments on page 764](#)
- [Specifying URL Attachments on page 765](#)

### 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, Event, 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.

When adding attachment files to Oracle CRM On Demand through Web services, it is better for performance reasons to send requests sequentially. For more information about best practices when using attachments, see ["Working with Attachments" on page 82](#).

### About The Attachment Element

In a Web services call, the content of an attachment must be contained in the <Attachment> element, which has the complex data type SiebelXMLAttachmentType, as shown in the following sample WSDL for an Attachment child object:

```
<xsd:complexType name="ListOfAttachment">
  <xsd:sequence>
    <xsd:element name="Attachment" maxOccurs="unbounded" minOccurs="0"
      type="xsdLocal1:Attachment" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Attachment">
  <xsd:sequence>
    <xsd:element name="Id" maxOccurs="1" minOccurs="0" type="xsdLocal1:string30"/>
    <xsd:element name="DisplayFileName" maxOccurs="1" minOccurs="0"
      type="xsdLocal1:string200" />
    <xsd:element name="FileNameOrURL" maxOccurs="1" minOccurs="0"
      type="xsdLocal1:string200" />
    <xsd:element name="FileExtension" maxOccurs="1" minOccurs="0"
      type="xsdLocal1:string10" />
    <xsd:element name="FileDate" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="FileSize" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="ContactId" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="ExternalSystemId" maxOccurs="1" minOccurs="0"
      type="xsdLocal1:string30" />
    <xsd:element name="AttachmentTag" maxOccurs="1" minOccurs="0" type="xsd:string"
      />
    <xsd:element name="Description" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="CreateDate" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="CreatedById" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="CreatedBy" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="ModId" maxOccurs="1" minOccurs="0" type="xsdLocal1:string30"/
    >
    <xsd:element name="ModifiedDate" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="ModifiedById" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="ModifiedBy" maxOccurs="1" minOccurs="0" type="xsd:string" />
    <xsd:element name="Attachment" maxOccurs="1" minOccurs="0"
      type="xsdLocal1:SibelXmlAttachmentType" />
  </xsd:sequence>
</xsd:complexType>
```

```

</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 762](#).
- **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">OM8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAA
.....
.....
AAAAAAAAAAAAAAAAAAAAAAAAA=</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>
          <NumberEmployees>300</NumberEmployees>
          <ListOfAttachment>
            <Attachment>
              <FileNameOrURL>Attached</FileNameOrURL>
              <FileExtension>doc</FileExtension>
              <DisplayFileName>To be reused</DisplayFileName>
              <Attachment
ContentID="reuse">OM8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/
CQAGAAAAAAAAAAAAACAAAA+AAAAAAAAAAAAEAAA+gAAAAEAAAD+////AAAAAPYAAD3AAAA////
////////////////////
```

```
..... The Base64 encoded stream goes here .....  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=
```

**</Attachment>**  

```
</Attachment>  
</ListOfAttachment>  
</Account>  
<Account>  
  <Description>This copying the attachment</Description>  
  <Location>San Mateo</Location>  
  <AccountName>Attach Test Copy</AccountName>  
  <NumberOfEmployees>300</NumberOfEmployees>  
  <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>
            <AttachmentTag>FILE</AttachmentTag>
            <Description></Description>
            <Attachment AttachmentIsTextData="false" Extension="xml "
ContentId="10A2-
RTX3">PD94bWwgdi VVRGLTgi Pz48P1NpZWJl bC10cm9wZXJ0eS1TZXQgRXNj YXBI
            <Base 64 encoded attachment data .....>
          </Attachment>
        </ListOfAttachment>
      </Account>
    </ListOfAccount>
  </ns:AccountWS_AccountQueryPage_Output>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

        </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>Attach 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 on page 767](#)

### 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 859](#) 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 147](#).

Table 859. 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 Activity History	CallActivityHistory
Call Product Detail	Call ProdDetail
Call Promotional Item	Call PromItemDrop
Call Sample Dropped	Call SampDrop
Campaign Recipient	ContactCampaign
Certification Request	CertificationRequest
Contact Address	ContactAddress

Table 859. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Contact Best Times	ContactBestTimes
Contact Interests	ContactInterest
Contact Relationship	ContactRelationship
Contact State License	CRMOD_LS_ContactLicenses
Course Enrollment	CourseEnrollment
Custom Object 01 Team	CustObj1 Team
Custom Object 02 Team	CustObj2 Team
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
Event	MedEdEvent
Financial Account	FinancialAccount
Financial Account Holder	FinancialAccountHolder
Financial Account Holding	FinancialAccountHolding
Financial Plan	FinancialPlan
Financial Product	FinancialProduct
Financial Transaction	FinancialTransaction
Household Team	HouseholdTeam
HCP Contact Allocation	HCPContactAllocation
Inventory Audit Report	CRMODLS_InventoryAuditReport
Inventory Period	CRMODLS_InventoryPeriod
Insurance Property	InsuranceProperty
Invitee	MedEdInvitee
Involved Party	InvolvedParty
MDF Request	MDFRequest
MDF Request Team	MDFRequestTeam



Table 859. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Message Response	CRMODLS_PCD_MSGRSP
Messaging Plan	CRMOD_LS_MessagingPlan
Messaging Plan Item	CRMOD_LS_MsgPlanItem
Messaging Plan Item Relation	CRMOD_LS_MsgPlnRel
Modification Tracking	CRMODLS_ModificationLog
Objective	CRMODLS_OBJECTIVE
Opportunity Competitor	OpportunityCompetitor
Opportunity Partner	OpportunityPartner
Partner Account	PartnerAccount
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
Product Indication	ProductIndication
Program Membership	ProgramPartner
Related Disclaimer	CRMODLS_RELDISC
Sample Disclaimer	CRMODLS_SIGNDISC
Sample Inventory	CRMODLS_SampleInventory
Sample Lot	CRMODLS_SampleLot
Sample Transaction	CRMOD_LS_Transactions
Signature	CRMODLS_Signature
Social Media	SocialMedia
Social Profile	SocialProfile

Table 859. Mapping of Display Object Names to Service API Object Names

UI Name	Service API Name
Special Pricing Request	SPRequest
Special Pricing Request Line Item	SPRequestLineItem
Special Pricing Request Team	SPRequestTeam
Transaction Item	CRMOD_LS_TransactionItems
Vehicle Financial Information	Vehicle FinInfo
Vehicle Sales History	Vehicle SalesHist
Vehicle Service History	Vehicle ServHist

# C

## Web Services On Demand Limiters

This appendix contains the following topics:

- [Web Services On Demand Limiters on page 771](#)
- [Web Services Request Rate on page 772](#)

### Web Services On Demand Limiters

The Web Services interface resources of Oracle CRM On Demand 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 860](#) 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 772](#).

Table 860. 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 860. Web Services On Demand Limiters

Limitier 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 242  
call return value 242

### **AccessProfileReadAll method**

arguments 243  
call return value 243

### **AccessProfileUpsert method**

arguments 244  
call return value 244

### **account child object**

about 703  
fields 704  
pick map 704  
picklists 704  
read-only fields 704  
required fields 704  
user key fields 704

### **account contact child object**

about 704  
fields 705  
pick map 705  
picklists 706  
read-only fields 705  
required fields 705  
user key fields 705

### **account contact role child object**

user key fields 725

### **account object**

about 378, 519  
child components 378  
fields 379, 521  
filterable fields 390  
methods 378, 521  
pick map fields 385, 521  
picklists 522  
read-only fields 379, 521  
required fields 379, 521  
status key 382  
user key fields 521

### **accreditation object**

about 582  
fields 582  
list of values 583  
methods 582  
pick map fields 583  
read-only fields 582

required fields 582

user key fields 583

### **accreditation request object**

about 584  
fields 584  
list of values 585  
methods 584  
pick map fields 585  
read-only fields 584  
required fields 584  
user key fields 585

### **ActionBarCustomWebAppletRead method**

arguments 245  
call return value 245

### **ActionBarCustomWebAppletReadAll method**

arguments 246  
call return value 246

### **ActionBarCustomWebAppletUpsert method**

arguments 247  
call return value 247

### **ActionBarLayoutReadAll method**

arguments 248  
call return value 248

### **ActionBarLayoutUpsert method**

arguments 250  
call return value 250

### **activity child object**

about 707  
fields 708  
pick map 708  
read-only fields 708  
required fields 708  
user key fields 709

### **activity object**

about 397, 522  
child components 398  
fields 399, 525  
filterable fields 403  
list of values 405, 527  
methods 398, 525  
parents 397  
pick map fields 402, 526  
picklists 396  
read-only fields 399, 525  
required fields 399, 525  
status key 401  
usage 397, 522

- user key fields 403, 527
- address child object** 709
  - fields 712
  - pick map 712
  - read-only fields 712
  - required fields 712
  - user key fields 712
- address fields** 32
- address location object**
  - about 539
  - methods 539
  - pick map fields 540
  - user keys 540
- Administrative Services API calls**
  - AccessProfileRead 241
  - AccessProfileReadAll 242
  - AccessProfileUpsert 243
  - ActionBarCustomWebAppletRead 244
  - ActionBarCustomWebAppletReadAll 245
  - ActionBarCustomWebAppletUpsert 246
  - ActionBarLayoutReadAll 247
  - ActionBarLayoutUpsert 248
  - AssignmentRuleGroupRead 253
  - AssignmentRuleGroupReadAll 254
  - AssignmentRuleGroupUpsert 254
  - AssignmentRuleRead 250
  - AssignmentRuleReadAll 251
  - AssignmentRuleUpsert 251
  - CascadingPicklistRead 256
  - CascadingPicklistReadAll 257
  - CascadingPicklistUpsert 257
  - ConcatenatedFieldRead 258
  - ConcatenatedFieldReadAll 259
  - ConcatenatedFieldUpsert 260
  - CurrencyRead 261
  - CurrencyReadAll 262
  - CurrencyUpsert 262
  - CurrentOrganizationRead 264
  - CustomHTMLHeadTagRead 269
  - CustomHTMLHeadTagUpsert 270
  - CustomRecordTypeRead 270
  - CustomRecordTypeReadAll 271
  - CustomRecordTypeUpsert 272
  - CustomWebLinkRead 274
  - CustomWebLinkReadAll 275
  - CustomWebLinkUpsert 275
  - CustomWebTabDelete 277
  - CustomWebTabRead 278
  - CustomWebTabReadAll 278
  - CustomWebTabUpsert 279
  - DetailPageCustomWebAppletRead 280
  - DetailPageCustomWebAppletReadAll 281
  - DetailPageCustomWebAppletUpsert 282
  - DynamicLayoutRead 283
  - DynamicLayoutReadAll 284
  - DynamicLayoutUpsert 285
  - EnableModificationTrackingRead 286
  - EnableModificationTrackingReadAll 287
  - EnableModificationTrackingUpsert 287
  - FieldAuditSetupReadAll 289
  - FieldAuditSetupRead 288
  - FieldAuditSetupUpsert 289
  - FieldManagementCreate 290
  - FieldManagementRead 291
  - FieldManagementReadAll 292
  - FieldManagementUpsert 293
  - HomepageCustomWebAppletRead 297
  - HomepageCustomWebAppletReadAll 298
  - HomepageCustomWebAppletUpsert 299
  - HomepageLayoutReadAll 295
  - HomeTabCustomWebAppletRead 301
  - HomeTabCustomWebAppletReadAll 301
  - HomeTabCustomWebAppletUpsert 302
  - HomepageLayoutUpsert 296
  - IndustryRead 304
  - IndustryReadAll 304
  - IntegrationEventQueue 305, 306, 307
  - ListAccessAndOrderRead 312
  - ListAccessAndOrderReadAll 313
  - ListAccessAndOrderUpsert 314
  - ListRead 309
  - ListReadAll 312
  - overview 233
  - PageLayoutFieldRead 315
  - PageLayoutFieldReadAll 315
  - PageLayoutFieldUpsert 316
  - PageLayoutRelatedInformationRead 317
  - PageLayoutRelatedInformationReadAll 318, 332
  - PageLayoutRelatedInformationUpsert 319
  - PageLayoutSectionReadAll 320
  - PageLayoutSectionUpsert 321
  - PicklistRead 322
  - PicklistReadAll 323
  - PicklistUpsert 324
  - PicklistValueGroupRead 325
  - PicklistValueGroupReadAll 326
  - PicklistValueGroupUpsert 327
  - ProcessAdministrationAll 328
  - ProcessAdministrationRead 328
  - ProcessAdministrationUpsert 329
  - RelatedInformationLayoutRead 331
  - RelatedInformationLayoutReadAll 332
  - RelatedInformationLayoutUpsert 333
  - ReportFolderRead 334
  - ReportFolderReadAll 335
  - ReportFolderUpsert 336
  - RoleRead 338



- RoleReadAll 339
- RoleUpsert 339
- SalesAssessmentTemplateRead 341
- SalesAssessmentTemplateReadAll 342
- SalesAssessmentTemplateUpsert 343
- SalesCategoryRead 345
- SalesCategoryReadAll 345
- SalesCategoryUpsert 346
- SearchLayoutRead 347
- SearchLayoutReadAll 348
- SearchLayoutUpsert 348
- SSOTokenRead 351
- WorkflowActionInsertOrUpdate 359
- WorkflowActionSetRead 352
- WorkflowActionSetReadForRuleSet 353
- WorkflowActionSetUpsert 355
- WorkflowRuleInsertOrUpdate 365
- WorkflowRuleSetDelete 360
- WorkflowRuleSetRead 361
- WorkflowRuleSetReadAll 362
- WorkflowRuleUpsert 363
- allocation object**
  - about 615
  - fields 615
  - list of values 616
  - methods 615
  - pick map fields 616
  - read-only fields 615
  - required fields 615
  - user key fields 616
- allotment usage object**
  - about 540
  - methods 541
  - user keys 541
- API calls**
  - AccessProfileRead 241
  - AccessProfileReadAll 242
  - AccessProfileUpsert 243
  - ActionBarCustomWebAppletRead 244
  - ActionBarCustomWebAppletReadAll 245
  - ActionBarCustomWebAppletUpsert 246
  - ActionBarLayoutReadAll 247
  - ActionBarLayoutUpsert 248
  - AssignmentRuleGroupRead 253
  - AssignmentRuleGroupReadAll 254
  - AssignmentRuleGroupUpsert 254
  - AssignmentRuleRead 250
  - AssignmentRuleReadAll 251
  - AssignmentRuleUpsert 251
  - AuditTrailQueryPage method 149
  - AuthAuthorizationUser method 155
  - BulkOpImportCreateRequest 366
  - BulkOpImportGetRequestDetail 370
  - BulkOpImportSendData 369
  - CascadingPicklistRead 256
  - CascadingPicklistReadAll 257
  - CascadingPicklistUpsert 257
  - ConcatenatedFieldRead 258
  - ConcatenatedFieldReadAll 259
  - ConcatenatedFieldUpsert 260
  - core methods 90
  - CurrencyRead 261
  - CurrencyReadAll 262
  - CurrencyUpsert 262
  - CurrentOrganizationRead 264
  - CustomHTMLHeadTagRead 269
  - CustomHTMLHeadTagUpsert 270
  - CustomRecordTypeRead 270
  - CustomRecordTypeReadAll 271
  - CustomRecordTypeUpsert 272
  - CustomWebLinkRead 274
  - CustomWebLinkReadAll 275
  - CustomWebLinkUpsert 275
  - CustomWebTabDelete 277
  - CustomWebTabRead 278
  - CustomWebTabReadAll 278
  - CustomWebTabUpsert 279
  - Delete method (Web Services v1.0) 91
  - Delete method (Web Services v2.0) 112
  - DeleteChild method 92
  - DeletedItemQueryPage method 157
  - DeleteEvents method 163
  - designing client applications 82
  - DetailPageCustomWebAppletRead 280
  - DetailPageCustomWebAppletReadAll 281
  - DetailPageCustomWebAppletUpsert 282
  - DynamicLayoutRead 283
  - DynamicLayoutReadAll 284
  - DynamicLayoutUpsert 285
  - Echo Argument 91, 108
  - EnableModificationTrackingRead 286
  - EnableModificationTrackingReadAll 287
  - EnableModificationTrackingUpsert 287
  - Execute method 116
  - ExportWebServiceCreateExportRequest 166
  - ExportWebServiceMonitorExportRequest 169
  - ExportWebServiceRetrieveExportAttachment 171
  - FieldAuditSetupRead 288
  - FieldAuditSetupReadAll 289
  - FieldAuditSetupUpsert 289
  - FieldManagementCreate 290
  - FieldManagementRead 291
  - FieldManagementReadAll 292
  - FieldManagementUpsert 293
  - GetEvents method 173
  - GetLists method 182

- GetMapping method 185
  - GetPicklistValues method 189
  - GetServerTime method 192
  - HomepageCustomWebAppletRead 297
  - HomepageCustomWebAppletReadAll 298
  - HomepageCustomWebAppletUpsert 299
  - HomepageLayoutReadAll 295
  - HomepageLayoutUpsert 296
  - HomeTabCustomWebAppletRead 301
  - HomeTabCustomWebAppletReadAll 301
  - HomeTabCustomWebAppletUpsert 302
  - IndustryRead 304
  - IndustryReadAll 304
  - Insert method (Web Services v1.0) 93
  - Insert method (Web Services v2.0) 120
  - InsertChild method 94
  - InsertOrUpdate method 95
  - IntegrationEventQueue 305, 306, 307
  - LeadConvert method 194
  - ListAccessAndOrderRead 312
  - ListAccessAndOrderReadAll 313
  - ListAccessAndOrderUpsert 314
  - ListRead 309
  - ListReadAll 312
  - LoginHistoryQueryPage method 200
  - LOVLanguageMode argument 109
  - MergeRecords method 203
  - MetadataChangeSummaryQueryPage method 207
  - PageLayoutFieldRead 315
  - PageLayoutFieldReadAll 315
  - PageLayoutFieldUpsert 316
  - PageLayoutRelatedInformationRead 317
  - PageLayoutRelatedInformationReadAll 318, 332
  - PageLayoutRelatedInformationUpsert 319
  - PageLayoutSectionReadAll 320
  - PageLayoutSectionUpsert 321
  - PicklistRead 322
  - PicklistReadAll 323
  - PicklistUpsert 324
  - PicklistValueGroupRead 325
  - PicklistValueGroupReadAll 326
  - PicklistValueGroupUpsert 327
  - ProcessAdministrationAll 328
  - ProcessAdministrationRead 328
  - ProcessAdministrationUpsert 329
  - QueryPage method (Web Services v1.0) 96
  - QueryPage method (Web Services v2.0) 123
  - RelatedInformationLayoutRead 331
  - RelatedInformationLayoutReadAll 332
  - RelatedInformationLayoutUpsert 333
  - replace operation 110
  - ReportFolderRead 334
  - ReportFolderReadAll 335
  - ReportFolderUpsert 336
  - RoleRead 338
  - RoleReadAll 339
  - RoleUpsert 339
  - SalesAssessmentTemplateRead 341
  - SalesAssessmentTemplateReadAll 342
  - SalesAssessmentTemplateUpsert 343
  - SalesCategoryRead 345
  - SalesCategoryReadAll 345
  - SalesCategoryUpsert 346
  - SalesProcessQueryPage method 210
  - SearchLayoutRead 347
  - SearchLayoutReadAll 348
  - SearchLayoutUpsert 348
  - Service API calls 147
  - SetPassword method 216
  - SetSessionTimeZone method 218
  - SSOTokenRead 351
  - Update method (Web Services v1.0) 106
  - Update method (Web Services v2.0) 145
  - UpdateCascadingPicklists method 224
  - UpdateChild method 107
  - UpdatePicklist method 220
  - UserUsageQueryPage method 229
  - ViewMode argument 109
  - WorkflowActionInsertOrUpdate 359
  - WorkflowActionSetRead 352
  - WorkflowActionSetReadForRuleSet 353
  - WorkflowActionSetUpsert 355
  - WorkflowRuleInsertOrUpdate 365
  - WorkflowRuleSetDelete 360
  - WorkflowRuleSetRead 361
  - WorkflowRuleSetReadAll 362
  - WorkflowRuleUpsert 363
- application object**
- about 585
  - fields 586
  - list of values 587
  - methods 586
  - pick map fields 586
  - read-only fields 586
  - required fields 586
  - user key fields 587
- arguments**
- Echo method 91, 108
  - LOVLanguageMode method 109
  - ViewMode method 109
- asset child object**
- fields 713
  - pick map 713
  - read-only fields 713
  - required fields 713
  - user key fields 713

**asset object**

- about 406, 541
- fields 407, 542
- filterable fields 408
- list of values 408, 543
- methods 407, 541
- pick map fields 408, 542
- read-only fields 407, 542
- required fields 407, 542
- status key 407
- usage 406, 541
- user key fields 408, 542

**AssignmentRuleGroupRead method**

- arguments 253
- call return value 253

**AssignmentRuleGroupReadAll method**

- arguments 254
- call return value 254

**AssignmentRuleGroupUpsert method**

- arguments 255
- call return value 256

**AssignmentRuleRead method**

- arguments 250
- call return value 250

**AssignmentRuleReadAll method**

- arguments 251
- call return value 251

**AssignmentRuleUpsert method**

- arguments 253
- call return value 253

**attachment child object**

- about 714
- fields 714
- read-only fields 714
- required fields 714
- user key fields 714

**attachment object 512****attachments 757****attendee call child object**

- about 715
- fields 715
- pick map 715
- read-only fields 715
- required fields 715
- user key fields 717, 720

**audit fields 375****audit records, lifetime 771****audit trail object 512****AuditTrailQueryPage method**

- about 149
- arguments 149
- call return value 150
- sample SOAP request 150

**AuthAuthorizationUser method**

- about 155
- arguments 155
- call return value 156
- sample SOAP request 156

**B****batch processing 80****best practices**

- API calls 82
- attachments 82
- batch processing 80
- error handling and logging 70
- flushing of caches 67
- indexed custom fields 85
- integration design 67
- integration management 86
- integration performance 80
- logging SOAP requests and responses 71
- login history 83
- moving customers between pods 86
- No Match Row Id value 84
- opportunity product revenue records 70
- outages 86
- outages and failures 71
- QueryPage method 83
- service allotments 79
- session management 81
- setting and querying blank values 70
- SSL certificates 86
- stateless Web services 68
- telephone numbers in Web services requests 76
- user usage 83
- Web services proxy classes 72
- Web services requests 67
- Web services transactions 72

**blocked product object**

- about 616
- fields 617
- methods 617
- pick map fields 617
- read-only fields 617
- required fields 617
- user key fields 618

**book child object**

- about 717
- fields 718
- pick map 718
- read-only fields 718
- required fields 718
- user key fields 718

**book object**

- about 409, 543

- child components 409
  - fields 410, 544
  - filterable fields 411
  - list of values 411, 544
  - methods 409, 543
  - pick map 411, 544
  - read-only fields 410, 544
  - required fields 410, 544
  - status key 410
  - user key fields 544
  - book user child object**
    - about 718
    - fields 719
    - pick map 719
    - read-only fields 719
    - required fields 719
    - user key fields 719
  - book user object** 512
  - BulkOpImportCreateRequest method**
    - arguments 368
    - call return value 369
  - BulkOpImportGetRequestDetail method**
    - arguments 371
    - call return value 371
  - BulkOpImportSendData method**
    - arguments 370
    - call return value 370
  - business plan object**
    - about 618
    - fields 619
    - list of values 620
    - methods 618
    - pick map fields 619
    - read-only fields 619
    - required fields 619
    - user key fields 619
- C**
- call activity history object**
    - about 620
    - fields 621
    - methods 620
    - user key fields 621
  - call product detail object**
    - about 621
    - fields 622
    - list of values 622
    - methods 621
    - pick map fields 622
    - read-only fields 622
    - required fields 622
    - user key fields 622
  - campaign child object**
    - about 720
    - fields 720
    - pick map 720
    - picklists 721
    - read-only fields 720
    - required fields 720
    - user key fields 721
  - campaign object**
    - about 412, 544
    - child components 412
    - fields 413, 545
    - filterable fields 419
    - list of values 420, 546
    - methods 412, 545
    - pick map field 416, 545
    - read-only fields 413, 545
    - required fields 413, 545
    - status key 415
    - user key fields 546
  - CascadingPicklistRead method**
    - arguments 256
    - call return value 256
  - CascadingPicklistReadAll method**
    - arguments 257
    - call return value 257
  - CascadingPicklistUpsert method**
    - arguments 258
    - call return value 258
  - category object**
    - about 546
    - methods 546
    - pick map field 547
    - read-only fields 547
    - required fields 547
    - user keys 547
  - certification object**
    - about 587
    - fields 588
    - list of values 589
    - methods 588
    - pick map fields 588
    - read-only fields 588
    - required fields 588
    - user key fields 589
  - certification request object**
    - about 589
    - fields 590
    - list of values 591
    - methods 590
    - pick map fields 590
    - read-only fields 590
    - required fields 590
    - user key fields 591
  - child components**

- account 378
- activity 398
- book 409
- campaign 412
- contact 421
- current user 440
- CustomObject1 442
- Event 492
- household 495
- lead 451
- opportunity 460
- portfolio 500
- service request 477
- solution 481
- user group 489
- vehicle 507
- child object**
  - user key fields 754
- child objects**
  - account 703
  - account contact 704
  - activity 707
  - address 709
  - asset 712
  - attachment 512, 714
  - attendee call 715
  - audit trail 512
  - book 717
  - book user 512, 718
  - campaign 720
  - competitor 512, 721
  - contact 722
  - contact role 724
  - custom object 725, 726
  - Custom Object 1 727
  - custom Web applet 728
  - delegated user 513, 729
  - household team 513
  - image 730
  - interests 514, 731
  - invitee 514, 732
  - lead 734
  - login history 514
  - multiple contact roles 514
  - note 735
  - opportunity 706, 736
  - opportunity child 737
  - opportunity team 514
  - portfolio team 515
  - products detailed 515, 737
  - promotional items dropped 515, 739
  - quota 516
  - recipient 516, 740
  - related account 516
  - related contact 516, 741
  - revenue 516, 742
  - sales assessment 744
  - sales assessment value 745
  - sample dropped 517, 745
  - service request 747
  - shared address 517, 748
  - sub book 517
  - subbook 752
  - team 518, 753
  - user 754
  - user default book per type 755
- claim object**
  - about 671
  - fields 672
  - methods 671
  - pick map fields 672
  - picklists 672
  - read-only fields 672
  - user key fields 672
- client applications, designing**
  - API calls 82
  - batch processing 80
  - error handling 70
  - error logging 70
  - handling outages 86
  - Insert method 82
  - InsertorUpdate method 82
  - session management 81
  - session pooling 81
  - Update method 82
- client integrations**
  - examples 18
- client name identification** 41
- client-side session management** 59
- company-specific WSDL** 26
- competitor child object**
  - about 721
  - fields 721
  - pick map 722
  - picklists 722
  - read-only fields 721
  - required fields 721
  - user key fields 722
- competitor object**
  - about 512
  - fields 513
- concatenated fields** 39
- ConcatenatedFieldRead method**
  - arguments 259
  - call return value 259
- ConcatenatedFieldReadAll method**
  - arguments 259
  - call return value 260

- ConcatenatedFieldUpsert method**
  - arguments 261
  - call return value 261
- contact best times object**
  - about 623
  - fields 623
  - list of values 624
  - methods 623
  - pick map fields 624
  - read-only fields 623
  - required fields 623
  - user key fields 624
- contact child object**
  - about 722
  - fields 723
  - pick map 723
  - read-only fields 723
  - required fields 723
  - user key fields 723
- contact object**
  - about 421, 547
  - child components 421
  - fields 422, 549
  - filterable fields 432
  - list of values 439, 550
  - methods 421, 549
  - objects not to use 440, 551
  - pick map 429, 550
  - read-only fields 422, 549
  - required fields 422, 549
  - status key 426
  - user key fields 550
- contact role child object**
  - about 724
  - fields 724
  - pick map 724
  - picklists 725
  - read-only fields 724
  - required fields 724
- contact state license object**
  - about 624
  - fields 625
  - list of values 626
  - methods 625
  - pick map 625
  - read-only fields 625
  - required fields 625
  - user key fields 626
- core methods**
  - Delete method (Web Services v1.0) 91
  - Delete method (Web Services v2.0) 112
  - DeleteChild method 92
  - Execute method 116
  - Insert method (Web Services v1.0) 93
  - Insert method (Web Services v2.0) 120
  - InsertChild method 94
  - InsertOrUpdate method 95
  - list of 90
  - MergeRecords 203
  - QueryPage method (Web Services v1.0) 96
  - QueryPage method (Web Services v2.0) 123
  - Update method (Web Services v1.0) 106
  - Update method (Web Services v2.0) 145
  - UpdateChild method 107
- core technologies 17**
  - about 17
- course enrollment object**
  - about 593
  - fields 594
  - list of values 595
  - methods 594
  - pick map fields 594
  - read-only fields 594
  - required fields 594
  - user key fields 595
- course object**
  - about 591
  - fields 592
  - list of values 593
  - methods 591
  - pick map fields 592
  - read-only fields 592
  - required fields 592
  - user key fields 593
- coverage object**
  - about 673
  - fields 674
  - methods 674
  - pick map fields 674
  - picklists 675
  - read-only fields 674
  - user key fields 675
- CRM On Demand**
  - custom fields 25
  - integration tag 26
- CRM On Demand Web Services 18**
  - communication with (figure) 18
  - filterable fields 377
  - parent and child relationships, about 25
  - reliability 21
  - Security 19
  - utilization 40
- currency and number fields 30**
- CurrencyRead method**
  - arguments 261
  - call return value 261
- CurrencyReadAll method**
  - arguments 262

- call return value 262
  - CurrencyUpsert method**
    - arguments 263
    - call return value 263
  - current user object**
    - about 440
    - child components 440
    - fields 441
    - filterable fields 441
    - methods 440
    - read-only fields 441
    - required fields 441
  - CurrentOrganizationRead method**
    - arguments 269
    - call return value 269
  - custom fields** 25
  - Custom Object 1 child object**
    - about 727
    - fields 727
    - read-only fields 727
    - required fields 727
    - user key fields 727
  - custom object child object**
    - about 725, 726
    - fields 725, 726
    - pick map 726
    - picklists 727
    - read-only fields 725, 726
    - required fields 725, 726
    - user key fields 726, 727
  - custom Web applet child object**
    - about 728
  - CustomHTMLHeadTagRead method**
    - arguments 269
    - call return value 269
  - CustomHTMLHeadTagUpsert method**
    - arguments 270
    - call return value 270
  - customization-specific WSDL** 26
  - CustomObject1 object**
    - about 442, 551
    - child components 442
    - fields 443, 553
    - filterable fields 449
    - list of values 451
    - methods 442, 553
    - pick map 447, 553
    - read-only fields 443, 553
    - required fields 443, 553
    - status key 445
    - user keys 449, 555
  - CustomRecordTypeRead method**
    - arguments 271
    - call return value 271
  - CustomRecordTypeReadAll method**
    - arguments 272
    - call return value 272
  - CustomRecordTypeUpsert method**
    - arguments 274
    - call return value 274
  - CustomWebLinkRead method**
    - arguments 274
    - call return value 275
  - CustomWebLinkReadAll method**
    - arguments 275
    - call return value 275
  - CustomWebLinkUpsert method**
    - arguments 277
    - call return value 277
  - CustomWebTabDelete method**
    - arguments 277
    - call return value 277
  - CustomWebTabRead method**
    - arguments 278
    - call return value 278
  - CustomWebTabReadAll method**
    - arguments 279
    - call return value 279
  - CustomWebTabUpsert method**
    - arguments 280
    - call return value 280
- D**
- damage object**
    - about 675
    - fields 676
    - methods 675
    - pick map fields 676
    - read-only fields 676
    - user key fields 676
  - Data Loader API calls**
    - available calls 366
    - BulkOpImportCreateRequest 366
    - BulkOpImportGetRequestDetail 370
    - BulkOpImportSendData 369
    - overview 366
  - date and time fields** 29
  - deal registration object**
    - about 595
    - fields 596
    - list of values 597
    - methods 596
    - pick map field 596
    - read-only fields 596
    - required fields 596
    - user keys 597
  - dealer object**

- about 504, 697
  - fields 505, 698
  - filterable fields 506
  - list of values 507, 699
  - methods 504, 698
  - pick map field 506, 698
  - read-only fields 505, 698
  - required fields 505, 698
  - status key 506
  - user keys 699
  - delegated user child object**
    - about 729
    - fields 729
    - picklists 729
    - read-only fields 729
    - required fields 729
    - user key fields 729
  - delegated user object** 513
  - Delete method**
    - arguments 92
  - Delete method (Web Services v1.0)**
    - usage 91
  - Delete method (Web Services v2.0)**
    - arguments 112
    - sample SOAP request 113
    - usage 112
  - DeleteChild method**
    - arguments 93
    - usage 92
  - DeletedItemQueryPage method**
    - about 157
    - arguments 160
    - call return value 161
    - sample SOAP request 161
  - DeleteEvents method**
    - about 163
    - call return value 164
    - sample SOAP request 165
  - DetailPageCustomWebAppletRead method**
    - arguments 281
    - call return value 281
  - DetailPageCustomWebAppletReadAll method**
    - arguments 282
  - DetailPageCustomWebAppletUpsert method**
    - arguments 283
    - call return value 283
  - division object**
    - about 555
    - fields 556
    - methods 556
    - pick map 556
    - read-only fields 556
    - required fields 556
  - user key fields 556
  - DynamicLayoutRead method**
    - arguments 284
    - call return value 284
  - DynamicLayoutReadAll method**
    - arguments 285
    - call return value 285
  - DynamicLayoutUpsert method**
    - arguments 286
    - call return value 286
- E
- Echo argument** 91, 108
  - email fields** 30
  - Enable Web Services Access privilege** 49
  - EnableModificationTrackingRead method**
    - call return value 286
  - EnableModificationTrackingReadAll method**
    - arguments 287
    - call return value 287
  - EnableModificationTrackingUpsert method**
    - arguments 288
    - call return value 288
  - error handling** 70
  - error logging** 70
  - establishing a session** 55
  - Event object**
    - about 492, 626
    - child components 492
    - fields 492, 627
    - filterable fields 494
    - list of values 494, 628
    - methods 492, 626
    - pick map fields 493, 627
    - read-only fields 492, 627
    - required fields 627
    - status key 493
    - user key fields 494, 627
  - exam object**
    - about 597
    - fields 598
    - list of values 599
    - methods 598
    - pick map fields 598
    - read-only fields 598
    - required fields 598
    - user key fields 599
  - exam registration object**
    - about 599
    - fields 600
    - list of values 601
    - methods 600
    - pick map fields 600



- read-only fields 600
  - required fields 600
  - user key fields 601
  - exchange rate history object**
    - about 729
    - fields 730
    - read-only fields 730
    - required fields 730
    - user key fields 730
  - exchange rate object**
    - about 557
    - fields 557
    - methods 557
    - read-only fields 557
    - required fields 557
    - user key fields 558
  - Execute method**
    - arguments 117
    - call return value 117
    - sample SOAP request 117
    - usage 116
  - ExportWebServiceCreateExportRequest method**
    - arguments 166
    - call return value 167
    - sample SOAP request 167
  - ExportWebServiceMonitorExportRequest method**
    - arguments 170
    - call return value 170
    - sample SOAP request 170
  - ExportWebServiceRetrieveExportAttachment method**
    - arguments 172
    - call return value 172
    - sample SOAP request 172
- F**
- features, what's new** 15
  - field types**
    - custom fields 25
    - supported 26
  - FieldAuditSetupRead method**
    - arguments 286, 288
    - call return value 288
  - FieldAuditSetupReadAll method**
    - arguments 289
    - call return value 289
  - FieldAuditSetupUpsert method**
    - arguments 290
    - call return value 290
  - FieldManagementCreate method**
    - arguments 291
    - call return value 291
  - FieldManagementRead method**
    - arguments 292
    - call return value 292
  - FieldManagementReadAll method**
    - arguments 293
    - call return value 293
  - FieldManagementUpsert method**
    - arguments 295
    - call return value 295
  - fields**
    - audit 375
    - custom fields 25
    - read-only, account child object 704
    - read-only, account contact child object 705
    - read-only, account object 379, 521
    - read-only, accreditation object 582
    - read-only, accreditation request object 584
    - read-only, activity child object 708
    - read-only, activity object 399, 525
    - read-only, address child object 712
    - read-only, allocation object 615
    - read-only, application object 586
    - read-only, asset child object 713
    - read-only, asset object 407, 542
    - read-only, attachment child object 714
    - read-only, attendee call child object 715
    - read-only, blocked product object 617
    - read-only, book child object 718
    - read-only, book object 410, 544
    - read-only, book user child object 719
    - read-only, business plan object 619
    - read-only, call product detail object 622
    - read-only, campaign child object 720
    - read-only, campaign object 413, 545
    - read-only, category object 547
    - read-only, certification object 588
    - read-only, certification request object 590
    - read-only, claim object 672
    - read-only, competitor child object 721
    - read-only, contact best times object 623
    - read-only, contact child object 723
    - read-only, contact object 422, 549
    - read-only, contact role child object 724
    - read-only, contact state license object 625
    - read-only, course enrollment object 594
    - read-only, course object 592
    - read-only, coverage object 674
    - read-only, current user object 441
    - read-only, Custom Object 1 child object 727
    - read-only, custom object child object 725, 726
    - read-only, CustomObject1 object 443, 553
    - read-only, damage object 676

- read-only, deal registration object 596
- read-only, dealer object 505, 698
- read-only, delegated user child object 729
- read-only, division object 556
- read-only, Event object 492, 627
- read-only, exam object 598
- read-only, exam registration object 600
- read-only, exchange rate history object 730
- read-only, exchange rate object 557
- read-only, financial account holder object 679
- read-only, financial account holding object 680
- read-only, financial account object 677
- read-only, financial plan object 682
- read-only, financial product object 683
- read-only, financial transaction object 685
- read-only, fund object 602
- read-only, group object 559
- read-only, HCP contact allocation object 628
- read-only, household object 496, 686
- read-only, image child object 731
- read-only, insurance property object 689
- read-only, interests child object 732
- read-only, inventory audit report object 631
- read-only, inventory period object 632
- read-only, invitee child object 733
- read-only, involved party object 690
- read-only, lead child object 734
- read-only, lead object 452, 560
- read-only, MDF request object 604
- read-only, message response object 634
- read-only, messaging plan item object 638
- read-only, messaging plan item relation object 640
- read-only, messaging plan object 636
- read-only, modification tracking object 563
- read-only, note child object 736
- read-only, note object 459, 564
- read-only, objective object 642
- read-only, opportunity child object 706, 737
- read-only, opportunity object 461, 565
- read-only, order item object 646
- read-only, order object 644
- read-only, partner object 606
- read-only, plan account object 647, 650
- read-only, plan contact object 649
- read-only, policy holder object 694
- read-only, policy object 692
- read-only, portfolio object 500, 696
- read-only, price list line item object 609
- read-only, price list object 608
- read-only, product category object 475
- read-only, product indication object 652
- read-only, product object 473, 568
- read-only, products detailed child object 738
- read-only, promotional items dropped child object 740
- read-only, recipient child object 741
- read-only, related contact child object 741
- read-only, related disclaimer object 653
- read-only, revenue child object 743
- read-only, sales assessment child object 744
- read-only, sales assessment value child object 745
- read-only, sample disclaimer object 656
- read-only, sample dropped child object 746
- read-only, sample inventory object 657
- read-only, sample lot object 660
- read-only, sample request item object 663
- read-only, sample request object 662
- read-only, sample transaction object 665
- read-only, service request child object 747
- read-only, service request object 478, 571
- read-only, shared address child object 751
- read-only, shared address object 538
- read-only, signature object 668
- read-only, social media object 573
- read-only, social profile object 575
- read-only, solution object 482, 576
- read-only, SP request object 612
- read-only, special pricing product object 611
- read-only, subbook child object 752
- read-only, team child object 753
- read-only, territory object 485, 578
- read-only, transaction item object 669
- read-only, user child object 754
- read-only, user default book per type child object 755
- read-only, user group object 489
- read-only, user object 487, 580
- read-only, vehicle object 508, 700
- required, account child object 704
- required, account contact child object 705
- required, account object 379, 521
- required, accreditation object 582
- required, accreditation request object 584
- required, activity child object 708
- required, activity object 399, 525
- required, address child object 712
- required, allocation object 615
- required, application object 586
- required, asset child object 713
- required, asset object 407, 542
- required, attachment child object 714
- required, attendee call child object 715
- required, blocked product object 617
- required, book child object 718

- required, book object 410, 544
- required, book user child object 719
- required, business plan object 619
- required, call product detail object 622
- required, campaign child object 720
- required, campaign object 413, 545
- required, category object 547
- required, certification object 588
- required, certification request object 590
- required, competitor child object 721
- required, contact best times object 623
- required, contact child object 723
- required, contact object 422, 549
- required, contact role child object 724
- required, contact state license object 625
- required, course enrollment object 594
- required, course object 592
- required, current user object 441
- required, Custom Object 1 child object 727
- required, custom object child object 725, 726
- required, CustomObject1 object 443, 553
- required, deal registration object 596
- required, dealer object 505, 698
- required, delegated user child object 729
- required, division object 556
- required, Event object 627
- required, exam object 598
- required, exam registration object 600
- required, exchange rate history object 730
- required, exchange rate object 557
- required, financial account holder object 679
- required, financial account holding object 680
- required, financial account object 677
- required, financial plan object 682
- required, financial product object 683
- required, financial transaction object 685
- required, fund object 602
- required, group object 559
- required, HCP contact allocation object 628
- required, household object 496, 686
- required, image child object 731
- required, interests child object 732
- required, inventory audit report object 631
- required, inventory period object 632
- required, invitee child object 733
- required, lead child object 734
- required, lead object 452, 560
- required, MDF request object 604
- required, message response object 634
- required, messaging plan item object 638
- required, messaging plan item relation object 640
- required, messaging plan object 636
- required, modification tracking object 563
- required, note child object 736
- required, note object 459, 564
- required, objective object 642
- required, opportunity child object 706, 737
- required, opportunity object 461, 565
- required, order item object 646
- required, order object 644
- required, plan account object 647
- required, plan contact object 649
- required, plan opportunity object 650
- required, price list line item object 609
- required, price list object 608
- required, product category object 475
- required, product indication object 652
- required, product object 473, 568
- required, products detailed child object 738
- required, promotional items dropped child object 740
- required, recipient child object 741
- required, related contact child object 741
- required, related disclaimer object 653
- required, revenue child object 743
- required, sales assessment child object 744
- required, sales assessment value child object 745
- required, sample disclaimer object 656
- required, sample dropped child object 746
- required, sample inventory object 657
- required, sample lot object 660
- required, sample request item object 663
- required, sample request object 662
- required, sample transaction object 665
- required, service request child object 747
- required, service request object 478, 571
- required, shared address child object 751
- required, shared address object 538
- required, signature object 668
- required, social media object 573
- required, social profile object 575
- required, solution object 482, 576
- required, SP request object 612
- required, special pricing product object 611
- required, subbook child object 752
- required, team child object 753
- required, territory object 485, 578
- required, transaction item object 669
- required, user child object 754
- required, user default book per type child object 755
- required, user group object 489
- required, user object 487, 580
- required, vehicle object 508, 700

**filterable fields** 377**financial account holder object**

about 678  
 fields 679  
 methods 678  
 pick map fields 679  
 picklists 679  
 read-only fields 679  
 required fields 679  
 user key fields 679

**financial account holding object**

about 680  
 fields 680  
 methods 680  
 pick map fields 681  
 read-only fields 680  
 required fields 680  
 user key fields 681

**financial account object**

about 676  
 fields 677  
 methods 677  
 pick map fields 677  
 picklists 678  
 read-only fields 677  
 required fields 677  
 user key fields 678

**financial plan object**

about 681  
 fields 682  
 methods 681  
 pick map fields 682  
 picklists 682  
 read-only fields 682  
 required fields 682  
 user key fields 682

**financial product object**

about 683  
 fields 683  
 methods 683  
 pick map fields 683  
 picklists 684  
 read-only fields 683  
 required fields 683  
 user key fields 684

**financial transaction object**

about 684  
 fields 685  
 methods 684  
 pick map fields 685  
 picklists 685  
 read-only fields 685  
 required fields 685  
 user key fields 685

**fund object**

about 601  
 fields 602  
 methods 602  
 pick map field 602  
 picklists 603  
 read-only fields 602  
 required fields 602  
 user key fields 602

**G****generating customized WSDL** 26**GetEvents method**

about 173  
 call return value 175  
 sample SOAP requests 176

**GetLists method**

about 182  
 arguments 183  
 call return value 183  
 sample SOAP request 183

**GetMapping method**

about 185  
 call return value 186  
 sample SOAP request 187

**GetPickList method**

sample SOAP request 191

**GetPicklistValues method**

about 189  
 arguments 190  
 call return value 190

**GetServerTime method**

about 192  
 call return value 193  
 sample SOAP request 193

**group object**

about 558  
 methods 558  
 pick map field 559  
 read-only fields 559  
 required fields 559  
 user key fields 559

**H****handling outages** 86**HCP contact allocation object**

about 628  
 fields 628  
 list of values 629  
 methods 628  
 pick map 629  
 read-only fields 628  
 required fields 628

- user key fields 629
  - HomepageCustomWebAppletRead method**
    - arguments 298
    - call return value 298
  - HomePageCustomWebAppletReadAll method**
    - call return value 282
  - HomepageCustomWebAppletReadAll method**
    - arguments 299
    - call return value 299
  - HomepageCustomWebAppletUpsert method**
    - arguments 300
    - call return value 300
  - HomepageLayoutReadAll method**
    - arguments 296
    - call return value 296
  - HomepageLayoutUpsert method**
    - arguments 297
    - call return value 297
  - HomeTabCustomWebAppletRead method**
    - arguments 301
    - call return value 301
  - HomeTabCustomWebAppletReadAll method**
    - arguments 302
    - call return value 302
  - HomeTabCustomWebAppletUpsert method**
    - arguments 303
    - call return value 304
  - household object**
    - about 495, 686
    - child components 495
    - fields 496, 686
    - filterable fields 498
    - list of values 499, 688
    - methods 495, 686
    - pick map field 498, 687
    - read-only fields 496, 686
    - required fields 496, 686
    - status key 497
    - user key fields 688
  - household team object** 513
- I**
- image child object**
    - about 730
    - fields 731
    - read-only fields 731
    - required fields 731
    - user key fields 731
  - Inbound Single Sign-On (SSO)** 65
  - indexed custom fields** 85
  - IndustryRead method**
    - arguments 304
    - call return value 304
  - IndustryReadAll method**
    - arguments 305
    - call return value 305
  - Insert method**
    - arguments 94
    - call return value 94
    - parent and child objects, effect on 94
  - Insert method (Web Services v1.0)**
    - usage 93
  - Insert method (Web Services v2.0)**
    - arguments 120
    - call return value 121
    - sample SOAP request 121
    - usage 120
  - InsertChild method**
    - arguments 95
    - usage 94
  - InsertOrUpdate method**
    - arguments 96
    - call return value 96
    - parent and child objects, effect on 95
    - usage 95
  - insurance property object**
    - about 688
    - fields 689
    - methods 688
    - pick map fields 689
    - picklists 689
    - read-only fields 689
    - user key fields 689
  - integration tag, viewing** 26
  - IntegrationEventQueue method**
    - arguments 306, 307, 309
    - call return value 306, 307, 309
  - integrations**
    - client integration examples 18
    - Web services session, integration requests 63
  - interests child object** 514
    - about 731
    - fields 732
    - pick map 732
    - picklists 732
    - read-only fields 732
    - required fields 732
    - user key fields 732
  - inventory audit report object**
    - about 630
    - fields 631
    - list of values 632
    - methods 630

- pick map field 631
  - read-only fields 631
  - required fields 631
  - user key fields 631
  - inventory period object**
    - about 632
    - fields 632
    - methods 632
    - pick map field 633
    - read-only fields 632
    - required fields 632
    - user key fields 633
  - invitee child object** 514
    - about 732
    - fields 733
    - pick map 733
    - picklists 733
    - read-only fields 733
    - required fields 733
    - user key fields 733
  - involved party object**
    - about 690
    - fields 690
    - methods 690
    - pick map fields 691
    - picklists 691
    - read-only fields 690
    - user key fields 691
  - isEncoded parameter** 61
- K**
- keys**
    - status key, about 375
    - status key, account object 382
    - status key, activity object 401
    - status key, asset object 407
    - status key, book object 410
    - status key, campaign object 415
    - status key, contact object 426
    - status key, CustomObject1 object 445
    - status key, dealer object 506
    - status key, Event object 493
    - status key, household object 497
    - status key, lead object 454
    - status key, note object 459
    - status key, opportunity object 464
    - status key, portfolio object 501
    - status key, product category object 476
    - status key, product object 473
    - status key, service request object 479
    - status key, solution object 483
    - status key, territory object 485
    - status key, user group object 490
    - status key, user object 488
    - status key, vehicle object 509
    - user key, about 374
    - user key, address location object 540
    - user key, allotment usage object 541
    - user key, category object 547
    - user key, period object 567
    - user key, price list line item object 610
    - user key, price list object 608
    - user key, product category object 476
    - user key, product indication object 652
    - user key, product object 474, 568
    - user key, related disclaimer object 654
    - user key, sample disclaimer object 656
    - user key, sample inventory object 658
    - user key, sample lot object 661
    - user key, sample transaction object 666
    - user key, service allotment object 571
    - user key, shared address object 539
    - user key, signature object 668
    - user key, territory object 578
    - user key, transaction item object 670
    - user key, user object 488, 581
- L**
- lead child object**
    - about 734
    - fields 734
    - pick map 734
    - read-only fields 734
    - required fields 734
    - user key fields 735
  - lead object**
    - about 451, 559
    - child components 451
    - fields 452, 560
    - methods 452, 560
    - pick map fields 455, 560
    - picklists 457, 561
    - read-only fields 452, 560
    - required fields 452, 560
    - status key 454
    - user key fields 561
  - LeadConvert method**
    - about 194
    - arguments 196
    - call return value 196
    - sample SOAP request 196
  - lifetime of audit records** 771
  - ListAccessAndOrderRead method**
    - arguments 313
    - call return value 313
  - ListAccessAndOrderReadAll method**

- arguments 313
- call return value 313
- ListAccessAndOrderUpsert method**
  - arguments 314
  - call return value 314
- ListRead method**
  - arguments 311
  - call return value 311
- ListReadAll method**
  - arguments 312
  - call return value 312
- locale-dependent access** 29
- logging in**
  - isEncoded parameter 61
  - Single Sign-On 57, 62
  - URL encoding 57, 60, 62
  - Web services session 60
- login history child object** 514
- LoginHistoryQueryPage method**
  - about 200
  - arguments 201
  - call return value 201
  - sample SOAP request 202
- LOVLanguageMode argument** 109

## M

- Manage Access to Web Services privilege** 49
- maskable fields** 39
- maximum number of records returned** 772
- maximum objects in a Web services request** 772
- maximum records returned** 772
- MDF request object**
  - about 603
  - fields 604
  - list of values 605
  - methods 603
  - pick map fields 604
  - read-only fields 604
  - required fields 604
  - user key fields 604
- MergeRecords method**
  - call return value 205
  - sample SOAP request 206
- message response object**
  - about 633
  - fields 634
  - list of values 635
  - methods 634
  - pick map fields 634
  - read-only fields 634
  - required fields 634
  - user key fields 635
- messaging plan item object**
  - about 637
  - fields 638
  - list of values 639
  - methods 638
  - pick map fields 638
  - read-only fields 638
  - required fields 638
  - user key fields 639
- messaging plan item relation object**
  - about 639
  - fields 640
  - list of values 641
  - methods 640
  - pick map fields 640
  - read-only fields 640
  - required fields 640
  - user key fields 641
- messaging plan object**
  - about 635
  - fields 636
  - list of values 637
  - methods 636
  - pick map fields 636
  - read-only fields 636
  - required fields 636
  - user key fields 637
- MetadataChangesSummaryQueryPage method**
  - sample SOAP request 208
- MetaDataChangeSummaryQueryPage method**
  - call return value 208
- MetadataChangeSummaryQueryPage method**
  - about 207
  - arguments 207
- methods called by**
  - account 378, 521
  - accreditation 582
  - accreditation request 584
  - activity 398, 525
  - address location 539
  - allocation 615
  - allotment usage 541
  - application 586
  - asset 407, 541
  - blocked product 617
  - book 409, 543
  - business plan 618
  - call activity history 620
  - call product detail 621
  - campaign 412, 545
  - category 546

- certification 588
  - certification request 590
  - claim 671
  - contact 421, 549
  - contact best times 623
  - contact state license 625
  - course 591
  - course enrollment 594
  - coverage 674
  - current user 440
  - CustomObject1 442, 553
  - damage 675
  - deal registration 596
  - dealer 504, 698
  - division 556
  - Event 492, 626
  - exam 598
  - exam registration 600
  - exchange rate 557
  - financial account 677
  - financial account holder 678
  - financial account holding 680
  - financial plan 681
  - financial product 683
  - financial transaction 684
  - fund 602
  - group 558
  - HCP contact allocation 628
  - household 495, 686
  - insurance property 688
  - inventory audit report 630
  - inventory period 632
  - involved party 690
  - lead 452, 560
  - MDF request 603
  - message response 634
  - messaging plan 636
  - messaging plan item 638
  - messaging plan item relation 640
  - modification tracking 562
  - note 458, 563
  - objective 641
  - opportunity 460, 565
  - order 644
  - order item 645
  - partner 605
  - period 567
  - plan account 647
  - plan contact 648
  - plan opportunity 650
  - policy 692
  - policy holder 694
  - portfolio 500, 695
  - price list 607
  - price list line item 609
  - product 473, 568
  - product category 475
  - product indication 651
  - related disclaimer 653
  - sample disclaimer 655
  - sample inventory 657
  - sample lot 659
  - sample request 661
  - sample request item 663
  - sample transaction 665
  - service allotment 570
  - service request 477, 571
  - shared address 538
  - signature 667
  - social media 573
  - social profile 575
  - solution 482, 576
  - SP request 612
  - special pricing product 610
  - territory 485, 578
  - transaction item 669
  - user 487, 580
  - user group 489
  - vehicle 508, 699
  - modification key**
    - about 375
  - modification tracking object**
    - about 562
    - fields 563
    - methods 562
    - read-only fields 563
    - required fields 563
    - user key fields 563
  - multiple contact roles child object** 514
  - multi-select picklists** 29
- N**
- NamedSearchSpec argument** 132
  - namespace, WSSE** 20
  - new features** 15
  - No Match Row Id value** 84
  - note child object**
    - about 735
    - fields 736
    - read-only fields 736
    - required fields 736
    - user key fields 736
  - note object**
    - about 458, 563
    - fields 459, 564
    - filterable fields 459
    - methods 458, 563



- read-only fields 459, 564
- required fields 459, 564
- status key 459
- user key fields 459, 564

**number and currency fields** 30

## O

**object display names** 767

**objective object**

- about 641
- fields 642
- list of values 643
- methods 641
- pick map fields 642
- read-only fields 642
- required fields 642
- user key fields 643

**objects**

- filterable fields 377
- modification key 375
- parent and child relationships, about 25
- pick maps 376
- status keys 375
- user keys 374

**objects, exposed**

See parent objects; child objects

**opportunity child object**

- about 706, 736, 737
- fields 706, 737
- pick map 707
- read-only fields 706, 737
- required fields 706, 737
- user key fields 707, 737

**opportunity object**

- about 459, 564
- child components 460
- fields 461, 565
- list of value fields 472, 566
- methods 460, 565
- pick map field 465, 565
- read-only fields 461, 565
- required fields 461, 565
- status key 464
- user key fields 566

**opportunity team child object** 514

**Oracle CRM On Demand Web Services**

- Oracle CRM On Demand parent and child record types 21
- pick maps 376
- status keys 375
- user keys 374

**Oracle CRM On Demand Web Services Toolkit**

- about 18

**Oracle On Demand Web Services Toolkit**

- CRM On Demand Web Services 18

**order item object**

- about 645
- fields 646
- list of values 646
- methods 645
- pick map fields 646
- read-only fields 646
- required fields 646
- user key fields 646

**order object**

- about 643
- fields 644
- methods 644
- pick map fields 644
- picklists 645
- read-only fields 644
- required fields 644
- user key fields 645

**Outbound Single Sign-On (SSO)**

- 65

**outbound SSO** 61, 66

## P

**PageLayoutFieldRead method**

- arguments 315
- call return value 315

**PageLayoutFieldReadAll method**

- arguments 316
- call return value 316

**PageLayoutFieldUpsert method**

- arguments 317
- call return value 317

**PageLayoutRelatedInformationRead method**

- arguments 318
- call return value 318

**PageLayoutRelatedInformationReadAll method**

- arguments 319
- call return value 319

**PageLayoutRelatedInformationUpsert method**

- arguments 320
- call return value 320

**PageLayoutSectionReadAll method**

- arguments 321
- call return value 321

**PageLayoutSectionUpsert method**

- arguments 322
- call return value 322

**parent and child objects**

- Insert method, effect on 94

- InsertOrUpdate method, effect on 95
- relationships, about 25
- Update method, effect of 106
- parent and child record types**
  - about and list of 21
- parent objects**
  - account 378, 519
  - accreditation 582
  - accreditation request 584
  - activity 397, 522
  - address 527, 539
  - allocation 615
  - allotment usage 540
  - application 585
  - asset 406, 541
  - blocked product 616
  - book 409, 543
  - business plan 618
  - call activity history 620
  - call product detail 621
  - campaign 412, 544
  - category 546
  - certification 587
  - certification request 589
  - claim 671
  - contact 547
  - contact best times 623
  - contact object 421
  - contact state license 624
  - course 591
  - course enrollment 593
  - coverage 673
  - current user 440
  - damage 675
  - deal registration 595
  - dealer 504, 697
  - division 555
  - Event 492, 626
  - exam 597
  - exam registration 599
  - exchange rate 557
  - exchange rate history 729
  - financial account 676
  - financial account holder 678
  - financial account holding 680
  - financial plan 681
  - financial product 683
  - financial transaction 684
  - fund 601
  - group 558
  - HCP contact allocation 628
  - household 495, 686
  - insurance property 688
  - inventory audit report 630
  - inventory period 632
  - involved party 690
  - lead 451, 559
  - MDF request 603
  - message response 633
  - messaging plan 635
  - messaging plan item 637
  - messaging plan item relation 639
  - modification tracking 562
  - note 458, 563
  - objective 641
  - opportunity 459, 564
  - order 643
  - order item 645
  - partner 605
  - period 566
  - plan account 646
  - plan contact 648
  - plan opportunity 650
  - policy 691
  - policy holder 693
  - portfolio 499, 695
  - price list 607
  - price list line item 608
  - product 472, 567, 569
  - product category 475
  - product indication 651
  - related disclaimer 652
  - sample disclaimer 654
  - sample inventory 657
  - sample lot 659
  - sample request 661
  - sample request item 663
  - sample transaction 664
  - service request 477, 571
  - signature 667
  - social media 573
  - social profile 574
  - solution 481, 576
  - SP request 611
  - special pricing product 610
  - territory 484, 577
  - transaction item 668
  - user 486, 579
  - user group 489
  - vehicle 507
- parent objects (Web Services v1.0)**
  - list of 377
- partner object**
  - about 605
  - fields 606
  - list of values 607
  - methods 605
  - pick map fields 606

- read-only fields 606
- user key fields 606
- period object**
  - about 566
  - methods 567
  - user keys 567
- pick map**
  - account child object 704
  - account contact child object 705
  - account object 385, 521
  - accreditation object 583
  - accreditation request object 585
  - activity child object 708
  - activity object 402, 526
  - address child object 712
  - address location object 540
  - allocation object 616
  - application object 586
  - asset child object 713
  - asset object 408, 542
  - attendee call child object 715
  - blocked product object 617
  - book child object 718
  - book object 411, 544
  - book user child object 719
  - business plan object 619
  - call product detail object 622
  - campaign child object 720
  - campaign object 416, 545
  - category object 547
  - certification object 588
  - certification request object 590
  - claim object 672
  - competitor child object 722
  - contact best times object 624
  - contact child object 723
  - contact CustomObject1 447, 553
  - contact object 429, 550
  - contact role child object 724
  - contact state license object 625
  - course enrollment object 594
  - course object 592
  - coverage object 674
  - custom object child object 726
  - damage object 676
  - deal registration object 596
  - dealer object 506, 698
  - division object 556
  - Event object 493, 627
  - exam object 598
  - exam registration object 600
  - financial account holder object 679
  - financial account holding object 681
  - financial account object 677
  - financial plan object 682
  - financial product object 683
  - financial transaction object 685
  - fund object 602
  - group object 559
  - HCP contact allocation object 629
  - household object 498, 687
  - insurance property object 689
  - interests child object 732
  - inventory audit report object 631
  - inventory period object 633
  - invitee child object 733
  - involved party object 691
  - lead child object 734
  - lead object pick map fields 455, 560
  - MDF request object 604
  - message response object 634
  - messaging plan item object 638
  - messaging plan item relation object 640
  - messaging plan object 636
  - objective object 642
  - opportunity child object 707
  - opportunity object 465, 565
  - Oracle CRM On Demand pick maps,
    - about 376
  - order item object 646
  - order object 644
  - partner object 606
  - plan account object 647
  - plan contact object 649
  - plan opportunity object 650
  - policy holder object 694
  - policy object 692
  - portfolio object 502, 696
  - price list line item object 609
  - price list object 608
  - product category object 476
  - product indication object 652
  - product object 473, 568
  - products detailed child object 738
  - promotional items dropped child object 740
  - related contact child object 742
  - revenue child object 743
  - sample disclaimer object 656
  - sample dropped child object 746
  - sample inventory object 658
  - sample lot object 660
  - sample request item object 664
  - sample request object 662
  - sample transaction object 666
  - service request child object 747
  - service request object 480, 572
  - shared address child object 751
  - shared address object 538

- signature object 668
- social media object 574
- social profile object 575
- SP request object 612
- special pricing product object 611
- subbook child object 752
- team child object 753
- territory object 486, 578
- transaction item object 669
- user child object 755
- user default book per type child object 756
- user group object 490
- user object 488, 580
- vehicle object 509, 700
- PicklistRead method**
  - arguments 323
  - call return value 323
- PicklistReadAll method**
  - arguments 323
  - call return value 324
- PicklistUpsert method**
  - arguments 325
  - call return value 325
- PicklistValueGroupRead method**
  - arguments 326
  - call return value 326
- PicklistValueGroupReadAll method**
  - arguments 326
  - call return value 327
- PicklistValueGroupUpsert method**
  - arguments 327
  - call return value 327
- plan account object**
  - about 646
  - fields 647
  - methods 647
  - pick map fields 647
  - read-only fields 647, 650
  - required fields 647
  - user key fields 648
- plan contact object**
  - about 648
  - fields 649
  - methods 648
  - pick map fields 649
  - read-only fields 649
  - required fields 649
  - user key fields 649
- plan opportunity object**
  - about 650
  - fields 650
  - methods 650
  - pick map fields 650
  - required fields 650
- user key fields 651
- policy holder object**
  - about 693
  - fields 694
  - methods 694
  - pick map fields 694
  - picklists 695
  - read-only fields 694
  - user key fields 695
- policy object**
  - about 691
  - fields 692
  - methods 692
  - pick map fields 692
  - picklists 693
  - read-only fields 692
  - user key fields 693
- portfolio object**
  - about 499, 695
  - child components 500
  - fields 500, 696
  - filterable fields 502
  - list of values 503, 697
  - methods 500, 695
  - pick map fields 502, 696
  - read-only fields 500, 696
  - status key 501
  - user key fields 502, 696
- portfolio team object**
  - about 515
- price list line item object**
  - about 608
  - fields 609
  - methods 609
  - pick map field 609
  - picklists 610
  - read-only fields 609
  - required fields 609
  - user keys 610
- price list object**
  - about 607
  - fields 608
  - methods 607
  - pick map field 608
  - picklists 608
  - read-only fields 608
  - required fields 608
  - user keys 608
- privileges**
  - Enable Web Services Access 49
  - Manage Access to Web Services. 49
- ProcessAdministrationRead method**
  - arguments 328
  - call return value 328

**ProcessAdministrationReadAll method**

arguments 329  
call return value 329

**ProcessAdministrationUpsert method**

arguments 330  
call return value 331

**product category object**

about 475  
fields 475, 547  
methods 475  
pick map field 476  
read-only fields 475  
required fields 475  
status key 476  
user keys 476

**product indication object**

about 651  
fields 652  
methods 651  
pick map fields 652  
read-only fields 652  
required fields 652  
user keys 652

**product object**

about 472, 567  
fields 473, 568  
methods 473, 568  
pick map field 473, 568  
picklists 474, 569  
read-only fields 473, 568  
required fields 473, 568  
status key 473  
user keys 474, 568

**products detailed child object**

about 737  
fields 738  
pick map 738  
read-only fields 738  
required fields 738  
user key fields 739

**products detailed object 515****promotional items dropped child object**

about 739  
fields 740  
pick map 740  
read-only fields 740  
required fields 740  
user key fields 740

**promotional items dropped object 515****Q****QueryPage method**

arguments 105

best practices 83

call return value 105

Oracle CRM On Demand query syntax,

about 99

querying data using Web Services 100

**QueryPage method (Web Services v1.0)**

about 96

**QueryPage method (Web Services v2.0)**

arguments 134  
call return value 135  
sample SOAP requests 135  
usage 123

**quota object 516****R****read-only fields**

account child object 704  
account contact child object 705  
account object 379, 521  
accreditation object 582  
accreditation request object 584  
activity child object 708  
activity object 399, 525  
address child object 712  
allocation object 615  
application object 586  
asset child object 713  
asset object 407, 542  
attachment child object 714  
attendee call child object 715  
blocked product object 617  
book child object 718  
book object 410, 544  
book user child object 719  
business plan object 619  
call product detail object 622  
campaign child object 720  
campaign object 413, 545  
category object 547  
certification object 588  
certification request object 590  
claim object 672  
competitor child object 721  
contact best times object 623  
contact child object 723  
contact object 422, 549  
contact role child object 724  
contact state license object 625  
course enrollment object 594  
course object 592  
coverage object 674  
current user object 441  
Custom Object 1 child object 727

- custom object child object 725, 726
- CustomObject1 object 443, 553
- damage object 676
- deal registration object 596
- dealer object 505, 698
- delegated user child object 729
- division object 556
- Event object 492, 627
- exam object 598
- exam registration object 600
- exchange rate history object 730
- exchange rate object 557
- financial account holder object 679
- financial account holding object 680
- financial account object 677
- financial plan object 682
- financial product object 683
- financial transaction object 685
- fund object 602
- group object 559
- HCP contact allocation object 628
- household object 496, 686
- image child object 731
- insurance property object 689
- interests child object 732
- inventory audit report object 631
- inventory period object 632
- invitee child object 733
- involved party object 690
- lead child object 734
- lead object 452, 560
- MDF request object 604
- message response object 634
- messaging plan item object 638
- messaging plan item relation object 640
- messaging plan object 636
- modification tracking object 563
- note child object 736
- note object 459, 564
- objective object 642
- opportunity child object 706, 737
- opportunity object 461, 565
- order item object 646
- order object 644
- partner object 606
- plan account object 647, 650
- plan contact object 649
- policy holder object 694
- policy object 692
- portfolio object 500, 696
- price list line item object 609
- price list object 608
- product category object 475
- product indication object 652
- product object 473, 568
- products detailed child object 738
- promotional items dropped child object 740
- recipient child object 741
- related contact child object 741
- related disclaimer object 653
- revenue child object 743
- sales assessment child object 744
- sales assessment value child object 745
- sample disclaimer object 656
- sample dropped child object 746
- sample inventory object 657
- sample lot object 660
- sample request item object 663
- sample request object 662
- sample transaction object 665
- service request child object 747
- service request object 478, 571
- shared address child object 751
- shared address object 538
- signature object 668
- social media object 573
- social profile object 575
- solution object 482, 576
- SP request object 612
- special pricing product object 611
- subbook child object 752
- team child object 753
- territory object 485, 578
- transaction item object 669
- user child object 754
- user default book per type child object 755
- user group object 489
- user object 487, 580
- vehicle object 508, 700
- recipient child object**
  - about 740
  - fields 741
  - read-only fields 741
  - required fields 741
  - user key fields 741
- recipient object** 516
- record ownership mode** 239
- record types**
  - Oracle CRM On Demand parent and child record types 21
- related account object** 516
- related contact child object**
  - about 741
  - fields 741
  - pick map 742
  - picklists 742
  - read-only fields 741
  - required fields 741

- user key fields 742
- related contact object** 516
- related disclaimer object**
  - about 652
  - fields 653
  - methods 653
  - picklists 654
  - read-only fields 653
  - required fields 653
  - user keys 654
- RelatedInformationLayoutRead method**
  - arguments 331
  - call return value 331
- RelatedInformationLayoutReadAll method**
  - arguments 332
  - call return value 332
- RelatedInformationLayoutUpsert method**
  - arguments 334
  - call return value 334
- release, what's new** 15
- reliability**
  - Web services reliability 21
- replace operation** 110
- ReportFolderRead method**
  - arguments 335
  - call return value 335
- ReportFolderReadAll method**
  - arguments 336
  - call return value 336
- ReportFolderUpsert method**
  - arguments 338
  - call return value 338
- request rate limit** 772
- request size limit** 772
- required fields**
  - account child object 704
  - account contact child object 705
  - account object 379, 521
  - accreditation object 582
  - accreditation request object 584
  - activity child object 708
  - activity object 399, 525
  - address child object 712
  - allocation object 615
  - application object 586
  - asset child object 713
  - asset object 407, 542
  - attachment child object 714
  - attendee call child object 715
  - blocked product object 617
  - book child object 718
  - book object 410, 544
  - book user child object 719
  - business plan object 619
  - call product detail object 622
  - campaign child object 720
  - campaign object 413, 545
  - category object 547
  - certification object 588
  - certification request object 590
  - competitor child object 721
  - contact best times object 623
  - contact child object 723
  - contact object 422, 549
  - contact role child object 724
  - contact state license object 625
  - course enrollment object 594
  - course object 592
  - current user object 441
  - Custom Object 1 child object 727
  - custom object child object 725, 726
  - CustomObject1 object 443, 553
  - deal registration object 596
  - dealer object 505, 698
  - delegated user child object 729
  - division object 556
  - Event object 627
  - exam object 598
  - exam registration object 600
  - exchange rate history object 730
  - exchange rate object 557
  - financial account holder object 679
  - financial account holding object 680
  - financial account object 677
  - financial plan object 682
  - financial product object 683
  - financial transaction object 685
  - fund object 602
  - group object 559
  - HCP contact allocation object 628
  - household object 496, 686
  - image child object 731
  - interests child object 732
  - inventory audit report object 631
  - inventory period object 632
  - invitee child object 733
  - lead child object 734
  - lead object 452, 560
  - MDF request object 604
  - message response object 634
  - messaging plan item object 638
  - messaging plan item relation object 640
  - messaging plan object 636
  - modification tracking object 563
  - note child object 736
  - note object 459, 564
  - objective object 642
  - opportunity child object 706, 737

- opportunity object 461, 565
- order item object 646
- order object 644
- plan account object 647
- plan contact object 649
- plan opportunity object 650
- price list line item object 609
- price list object 608
- product category object 475
- product indication object 652
- product object 473, 568
- products detailed child object 738
- promotional items dropped child object 740
- recipient child object 741
- related contact child object 741
- related disclaimer object 653
- revenue child object 743
- sales assessment child object 744
- sales assessment value child object 745
- sample disclaimer object 656
- sample dropped child object 746
- sample inventory object 657
- sample lot object 660
- sample request item object 663
- sample request object 662
- sample transaction object 665
- service request child object 747
- service request object 478, 571
- shared address child object 751
- shared address object 538
- signature object 668
- social media object 573
- social profile object 575
- solution object 482, 576
- SP request object 612
- special pricing product object 611
- subbook child object 752
- team child object 753
- territory object 485, 578
- transaction item object 669
- user child object 754
- user default book per type child object 755
- user group object 489
- user object 487, 580
- vehicle object 508, 700
- revenue child object**
  - about 742
  - fields 743
  - pick map 743
  - picklists 744
  - read-only fields 743
  - required fields 743
  - user key fields 743
- revenue object** 516
- RoleRead method**
  - arguments 338, 351
  - call return value 338, 352
- RoleReadAll method**
  - arguments 339
  - call return value 339
- RoleUpsert method**
  - arguments 341
  - call return value 341
- S**
- sales assessment child object** 744
  - fields 744
  - read-only fields 744
  - required fields 744
  - user key fields 744
- sales assessment value child object** 745
  - fields 745
  - read-only fields 745
  - required fields 745
  - user key fields 745
- SalesAssessmentTemplateRead method**
  - arguments 342
  - call return value 342
- SalesAssessmentTemplateReadAll method**
  - arguments 343
  - call return value 343
- SalesAssessmentTemplateUpsert method**
  - arguments 344
  - call return value 345
- SalesCategoryRead method**
  - arguments 345
  - call return value 345
- SalesCategoryReadAll method**
  - arguments 346
  - call return value 346
- SalesCategoryUpsert method**
  - arguments 347
  - call return value 347
- SalesProcessQueryPage method**
  - about 210
  - arguments 210
  - call return value 211
  - sample SOAP requests 212
- sample disclaimer object**
  - about 654
  - fields 656
  - methods 655
  - pick map field 656
  - picklists 657
  - read-only fields 656
  - required fields 656
  - user keys 656



**sample dropped child object**

- about 745
- fields 746
- pick map 746
- read-only fields 746
- required fields 746
- user key fields 746

**sample dropped object** 517**sample inventory object**

- about 657
- fields 657
- methods 657
- pick map field 658
- read-only fields 657
- required fields 657
- user keys 658

**sample lot object**

- about 659
- fields 660
- methods 659
- pick map field 660
- read-only fields 660
- required fields 660
- user keys 661

**sample request item object**

- about 663
- fields 663
- methods 663
- pick map fields 664
- picklists 664
- read-only fields 663
- required fields 663
- user key fields 664

**sample request object**

- about 661
- fields 662
- methods 661
- pick map fields 662
- picklists 662
- read-only fields 662
- required fields 662
- user key fields 662

**sample transaction object**

- about 664
- fields 665
- methods 665
- pick map field 666
- picklists 666
- read-only fields 665
- required fields 665
- user keys 666

**SearchLayoutRead method**

- arguments 347
- call return value 347

**SearchLayoutReadAll method**

- arguments 348
- call return value 348

**SearchLayoutUpsert method**

- arguments 351
- call return value 351

**security**

- Web services security 19

**server-side session management** 56**service allotment object**

- about 569
- methods 570
- user keys 571

**service allotments**

- best practices 79
- overview 45
- Web services concurrent request allotment 45
- Web services operations allotment 45

**Service API calls** 147

- AuditTrailQueryPage 149
- AuthAuthorizationUser 155
- DeletedItemQueryPage 157
- DeleteEvents 163
- ExportWebServiceCreateExportRequest 166
- ExportWebServiceMonitorExportRequest 169
- ExportWebServiceRetrieveExportAttachment 171
- GetEvents 173
- GetLists 182
- GetMapping 185
- GetPicklistValues 189
- GetServerTime 192
- LeadConvert 194
- LoginHistoryQueryPage 200
- MetadataChangeSummaryQueryPage 207
- SalesProcessQueryPage 210
- SetPassword 216
- SetSessionTimeZone 218
- UpdateCascadingPicklists 224
- UpdatePicklist 220
- UserUsageQueryPage 229

**service request child object**

- about 747
- fields 747
- pick map 747
- read-only fields 747
- required fields 747
- user key fields 748

**service request object**

- about 477, 571
- child components 477
- fields 478, 571

- filterable fields 480
- methods 477, 571
- pick map 480, 572
- picklists 481, 572
- read-only fields 478, 571
- required fields 478, 571
- status key 479
- user key fields 572
- session limit quota** 771
- session management** 55, 81
- session management, client-side** 59
- session management, server-side** 56
- session pooling** 81
- session time-outs** 81
- SetPassword** 216
- SetPassword method**
  - sample SOAP request 217
- SetSessionTimeZone method**
  - about 218
  - call return value 219
  - sample SOAP request 219
- setting up, Web services** 49
- setting up, Web services session**
  - establishing and managing 55
  - integration requests 63
  - limits 771
  - logging in 60
  - logging off 62
- shadow fields** 130
- shared address child object** 748
  - fields 751
  - pick map 751
  - picklists 751
  - read-only fields 751
  - required fields 751
  - user key fields 751
- shared address object** 517
  - about 527
  - fields 538
  - list of values 539
  - methods 538
  - pick map fields 538
  - read-only fields 538
  - required fields 538
  - user keys 539
- shared addresses, converting to** 528
- signature object**
  - about 667
  - fields 668
  - methods 667
  - pick map field 668
  - read-only fields 668
  - required fields 668
  - user keys 668
- Single Sign-On (SSO)**
  - 65
  - logging in using 57, 62
  - outbound SSO 61, 66
- social media object**
  - about 573
  - fields 573
  - methods 573
  - pick map 574
  - read-only fields 573
  - required fields 573
  - user key fields 574
- social profile object**
  - about 574
  - fields 575
  - methods 575
  - pick map 575
  - read-only fields 575
  - required fields 575
  - user key fields 576
- solution object**
  - about 481, 576
  - child components 481
  - fields 482, 576
  - filterable fields 483
  - list of value fields 484, 577
  - methods 482, 576
  - read-only fields 482, 576
  - required fields 482, 576
  - status key 483
  - user key fields 577
- SP request object**
  - about 611
  - fields 612
  - list of values 613
  - methods 612
  - pick map fields 612
  - read-only fields 612
  - required fields 612
  - user key fields 613
- special pricing product object**
  - about 610
  - fields 611
  - methods 610
  - pick map fields 611
  - read-only fields 611
  - required fields 611
  - user key fields 611
- stateful Web services** 50, 60, 81
- stateless Web services** 50, 56, 68
- status key**
  - about 375
  - account object 382
  - activity object 401

- asset object 407
  - book object 410
  - campaign object 415
  - contact object 426
  - CustomObject1 object 445
  - dealer object 506
  - Event object 493
  - household object 497
  - lead object 454
  - note object 459
  - opportunity object 464
  - portfolio object 501
  - product category object 476
  - product object 473
  - service request object 479
  - solution object 483
  - territory object 485
  - user group object 490
  - user object 488
  - vehicle object 509
  - sub book object** 517
  - subbook child object**
    - about 752
    - fields 752
    - pick map 752
    - read-only fields 752
    - required fields 752
    - user key fields 752
- T**
- team child object**
    - about 753
    - fields 753
    - pick map 753
    - picklists 754
    - read-only fields 753
    - required fields 753
  - team object** 518
  - territory object**
    - about 484, 577
    - fields 485, 578
    - methods 485, 578
    - pick map field 486, 578
    - picklists 486, 579
    - read-only fields 485, 578
    - required fields 485, 578
    - status key 485
    - user key 578
  - time and date fields** 29
  - transaction item object**
    - about 668
    - fields 669
    - methods 669
  - pick map field 669
  - read-only fields 669
  - required fields 669
  - user keys 670
- U**
- unshared addresses, converting from** 528
  - Update method**
    - arguments 106
    - call return value 107
    - parent and child objects, effect of 106
  - Update method (Web Services v1.0)**
    - usage 106
  - Update method (Web Services v2.0)**
    - arguments 145
    - call return value 146
    - sample SOAP request 146
    - usage 145
  - UpdateCascadingPicklists method**
    - about 224
    - arguments 225
    - call return value 226
    - sample SOAP request 226
    - sample SOAP requests 226
  - UpdateChild method**
    - arguments 107
    - usage 107
  - UpdatePicklist method**
    - arguments 221
    - call return value 221
    - sample SOAP requests 221
  - UpdatePicklistValues method**
    - about 220
  - URL parameterization** 86
  - usage object**
    - usage 486
  - user child object**
    - about 754
    - fields 754
    - pick map 755
    - read-only fields 754
    - required fields 754
    - user key fields 755
  - user default book per type child object**
    - about 755
    - fields 755
    - pick map 756
    - read-only fields 755
    - required fields 755
    - user key fields 756
  - user group object**
    - about 489
    - child components 489

fields 489, 508, 559, 700  
 filterable fields 491  
 methods 489  
 pick map field 490  
 read-only fields 489  
 required fields 489  
 status key 490

**user key**

Event object 627  
 inventory period object 633  
 message response object 635  
 messaging plan item object 639  
 messaging plan item relation object 641  
 messaging plan object 637  
 modification tracking object 563  
 objective object 643  
 order item object 646  
 order object 645  
 plan account object 648  
 plan contact object 649  
 plan opportunity object 651  
 sample request item object 664  
 sample request object 662

**user keys**

about 374  
 account object 521  
 accreditation object 583  
 accreditation request object 585  
 activity object 527  
 address location object 540  
 allocation object 616  
 allotment usage object 541  
 application object 587  
 asset object 542  
 blocked product object 618  
 book object 544  
 business plan object 619  
 call activity history object 621  
 call product detail object 622  
 campaign object 546  
 category object 547  
 certification object 589  
 certification request object 591  
 claim object 672  
 contact best times object 624  
 contact object 550  
 contact state license object 626  
 course enrollment object 595  
 course object 593  
 coverage object 675  
 CustomObject1 object 555  
 damage object 676  
 deal registration object 597  
 dealer object 699

division object 556  
 exam object 599  
 exam registration object 601  
 exchange rate history object 730  
 exchange rate object 558  
 financial account holder object 679  
 financial account holding object 681  
 financial account object 678  
 financial plan object 682  
 financial product object 684  
 financial transaction object 685  
 fund object 602  
 group object 559  
 HCP contact allocation object 629  
 household object 688  
 insurance property object 689  
 inventory audit report object 631  
 involved party object 691  
 lead object 561  
 MDF request object 604  
 note object 564  
 opportunity object 566  
 partner object 606  
 period object 567  
 policy holder object 695  
 policy object 693  
 portfolio object 696  
 price list line item object 610  
 price list object 608  
 product category object 476  
 product indication object 652  
 product object 474, 568  
 related disclaimer object 654  
 sample disclaimer object 656  
 sample inventory object 658  
 sample lot object 661  
 sample transaction object 666  
 service allotment object 571  
 service request object 572  
 shared address object 539  
 signature object 668  
 social media object 574  
 social profile object 576  
 solution object 577  
 SP request object 613  
 special pricing product object 611  
 territory object 578  
 transaction item object 670  
 user object 488, 581  
 vehicle object 700

**user object**

about 486, 579  
 fields 487, 580  
 methods 487, 580

- pick map field 488, 580
  - read-only fields 487, 580
  - required fields 487, 580
  - status key 488
  - usage 579
  - user keys 488, 581
  - UserNameToken profile** 20
  - UserUsageQueryPage method**
    - about 229
    - arguments 230
    - call return value 230
    - sample SOAP request 231
  - utilization**
    - Web services reliability 40
- V**
- vehicle object**
    - about 507
    - child components 507
    - filterable fields 509
    - methods 508, 699
    - pick map field 509, 700
    - picklists 510, 701
    - read-only fields 508, 700
    - required fields 508, 700
    - status key 509
    - user key fields 700
  - ViewMode argument** 109
- W**
- Web link fields** 36
  - Web service request rate** 771
  - Web service request size** 772
  - Web service session inactivity time limit** 772
  - Web services**
    - core technologies 17
    - custom fields 25
    - integration tag 26
    - reliability 21
    - security 19
    - utilization 40
  - Web services privilege**
    - Enable Web Services Access 49
    - Manage Access to Web Services 49
  - Web services proxy classes** 72
  - Web Services R16 Compatibility Mode setting** 50
  - Web services session**
    - establishing and managing 55
    - integration requests 63
    - logging in 60
    - logging off 62
    - maximum objects in Web Services request 772
    - maximum records returned 772
    - request rate limit 772
    - request size limit 772
  - Web Services Toolkit**
    - See Oracle CRM On Demand Web Services Toolkit
  - Web services transactions** 72
  - WorkflowActionSetRead method**
    - arguments 353
    - call return value 353
  - WorkflowActionSetReadReadForRuleSet method**
    - arguments 354
    - call return value 354
  - WorkflowActionSetUpsert method**
    - arguments 359, 360
    - call return value 359, 360
  - WorkflowRuleDelete method**
    - arguments 361
  - WorkflowRuleSetDelete method**
    - call return value 361
  - WorkflowRuleSetRead method**
    - arguments 361
    - call return value 362
  - WorkflowRuleSetReadAll method**
    - arguments 362
    - call return value 362
  - WorkflowRuleSetUpsert method**
    - arguments 364, 365
    - call return value 365, 366
  - WSDL, generating company-specific** 26
  - WS-I Basic Security Profile** 19, 20
  - WSSE namespace** 20, 50

