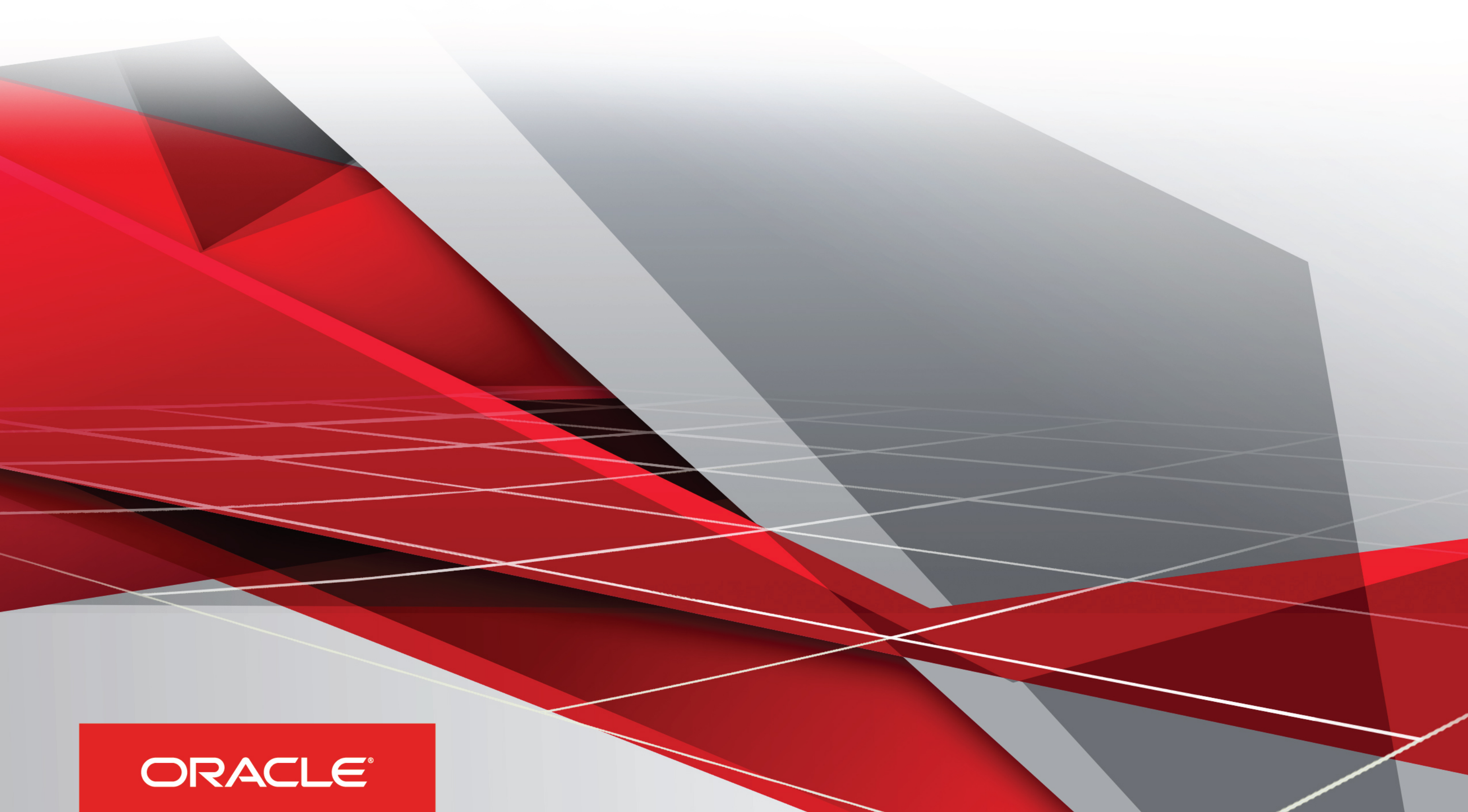
Oracle CRM On Demand

Desktop Administration Guide Version 5.2 Rev A

March 2019



Oracle CRM On Demand
Desktop Administration Guide Version 5.2 Rev A

March 2019

Part Number: F18392-01

Authors: Oracle CRM On Demand Information Development Team

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

Contents

Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at *http://docs.oracle.com/*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: *oracle_fusion_applications_help_ww_grp@oracle.com*.

1 What’s New in This Release

What’s New in Oracle CRM On Demand Desktop Administration Guide, Version 5.2 Revision A

The following information lists changes in this version of the documentation to support Version 5.2 of the software.

Topic	Description
Throughout this guide.	Modified topics. Removed references to Microsoft Outlook 2003 as this version of Microsoft Outlook is no longer supported. In addition, removed references to the forms_11.xml and toolbars_11.xml files as they are applicable only to Microsoft Outlook 2003.

What’s New in Oracle CRM On Demand Desktop Administration Guide, Version 5.2

The following information lists changes in this version of the documentation to support Version 5.2 of the software.

Topic	Description
Add-Ins Incompatible with Oracle CRM On Demand Desktop	Modified topic. The following have been added to the list of incompatible add-ins: • Evernote Clipper for Outlook • G Suite Sync for Microsoft Outlook (Google Apps Sync for Microsoft Outlook)

2 Overview of Oracle CRM On Demand Desktop

Overview of Oracle CRM On Demand Desktop

This chapter provides an overview of Oracle CRM On Demand Desktop. It includes the following topics:

- *About Using Oracle CRM On Demand Desktop*
- *Scenarios for Using Oracle CRM On Demand Desktop*

Oracle CRM On Demand Desktop is a client-side desktop application that is an extension to Microsoft Outlook. It allows the user to work with Oracle CRM On Demand in a native Microsoft Outlook environment. The data is accessible offline in Microsoft Outlook, and it can be synchronized with the Oracle CRM On Demand server when Oracle CRM On Demand Desktop is connected to the Internet. Oracle also supports an Oracle CRM On Demand Desktop Lite product. For more information on Oracle CRM On Demand Desktop Lite, see *Overview of Oracle CRM On Demand Desktop Lite*.

About Using Oracle CRM On Demand Desktop

Oracle CRM On Demand Desktop centralizes essential business information in the familiar Microsoft Outlook environment. This centralization complements the existing capabilities that Oracle CRM On Demand provides. These features help the user to perform common interactions. The user can perform the following actions with Oracle CRM On Demand Desktop:

- **Log in Using Single Sign-On (SSO).** Oracle CRM On Demand Desktop Single Sign-On (SSO) is a single sign-on feature that allows you to implement single sign-on for the Oracle CRM On Demand Desktop client. It supports standard Web browser functionality, such as HTTP forms, cookies, and process redirects. It also provides Oracle CRM On Demand Desktop users with a flexible sign-on system that supports Security Assertion Markup Language (SAML) versions 1.0 and 2.0.
- **Manage Oracle CRM On Demand data.** Users can manage Oracle CRM On Demand data such as accounts, contacts, leads, opportunities and activities, including appointments and tasks, directly in Microsoft Outlook.
- **Synchronize data.** Performs bidirectional, incremental synchronization between Oracle CRM On Demand Desktop on the Microsoft Outlook and Oracle CRM On Demand, which helps to keep the data in these applications up-to-date and consistent.
- **Work while disconnected.** Allows you to work with your records when you are disconnected from the Internet.

For your organization, some benefits include:

- Increased user adoption of your business processes and tools. Oracle CRM On Demand Desktop does not require the user to use an application that the user is not familiar with.
- Increased accessibility to data because the user is not required to log in to Oracle CRM On Demand to view and maintain Oracle CRM On Demand data.
- Decreased training costs. Most users are already familiar with Microsoft Outlook. You can focus your training on Oracle CRM On Demand processes instead of spending time and resources on learning how to perform simple interactions.

Scenarios for Using Oracle CRM On Demand Desktop

This topic describes several scenarios of how you can use Oracle CRM On Demand Desktop with Microsoft Outlook. It includes the following information:

- *Scenario for Working with an Activity That Is Associated with an Opportunity*
- *Scenario for Managing Contact Information*
- *Scenario for Managing Account Information*
- *Scenario for Associating an Email with an Opportunity*
- *Scenario for Managing an Opportunity*

Scenario for Working with an Activity That Is Associated with an Opportunity

This scenario gives one example of how you might use Oracle CRM On Demand Desktop with an activity that is associated with an opportunity. You might use Oracle CRM On Demand Desktop differently, depending on your business model.

On Friday afternoon, a sales manager reviews the Big Deal opportunity and realizes that it has been inactive for some time. The manager does the following work in Microsoft Outlook to assign the task to a sales representative:

- Creates a new Microsoft Outlook task
- Associates the Big Deal opportunity with the newly created Microsoft Outlook task, and completes the other details of the task
- Sends the task to the assigned sales representative

On Monday, the representative uses Microsoft Outlook to view opportunities. The representative examines the opportunity and notices the new activity, notices the assignor, and realizes that an upcoming demonstration must be revised. To examine more information the representative drills down on the activity record. On Thursday, the representative finishes revising the demonstration and changes the activity status to Done.

To view the activities that are assigned and completed, on Friday, the sales manager uses a customized version of Oracle CRM On Demand Desktop to choose the Oracle CRM On Demand Activities view in the Activities Folder.

Scenario for Managing Contact Information

This scenario gives one example of how you might use Oracle CRM On Demand Desktop to manage contact information. You might use Oracle CRM On Demand Desktop differently, depending on your business model.

A sales representative works at High-Tech Office Expo and manages many customers, including a customer named Company Y. While at High-Tech Office Expo, the sales representative meets a new contact who is the CEO of Company X, which is a competitor of Company Y. The sales representative also encounters an old college friend who just moved to town. They trade contact information.

The sales representative creates a new contact in Microsoft Outlook and enters information about the CEO in this new contact record. While creating this contact, the representative links the contact to the existing Company X account. Because the representative must share this contact with colleagues, the representative clicks the Sharing Bar. Oracle CRM On

Demand Desktop then marks the contact as shared and changes the Sharing Bar to an orange color, which indicates that the contact is shared.

In the same Contacts folder, the sales representative creates a new contact for the old college friend. Because the default option does not share the contact with Oracle CRM On Demand, the contact remains not shared.

The sales representative must call the VP of sales at Company Y, who is represented in Oracle CRM On Demand data as another contact. The representative finds the contact record and then locates the cell phone number for the contact. The contact record for the VP displays in Microsoft Outlook along with all the other contacts that the representative can normally view in Oracle CRM On Demand. In the meantime, another user at High-Tech Office Expo updates the contact information for the VP. When the sales representative synchronizes, Oracle CRM On Demand Desktop displays this updated information in Microsoft Outlook.

Scenario for Managing Account Information

This scenario gives one example of how you might use Oracle CRM On Demand Desktop to manage account information. You might use Oracle CRM On Demand Desktop differently, depending on your business model.

Company Z is one of the smaller accounts that a sales representative manages. Because Company Z recently relocated, the representative must update the address details that are associated with the account. The representative drills down on the account in the accounts view and then enters the new address in the form.

To make sure that everyone on the account team is aware of the new location, the representative must send an email to the account team. To include all account team members, the representative uses the Email to Account Team button in Microsoft Outlook. Oracle CRM On Demand Desktop enters email addresses for the entire account team in an email message and then displays the message. The representative types a brief note about the new location and then sends the email.

While the representative is in the account record for Company Z, the representative decides to add a new contact to the account. To do this, the representative clicks Add Contact in the Contacts section, and then Oracle CRM On Demand Desktop displays the Account Contacts SalesBook dialog box where the representative can select a contact for association or create a new contact.

Scenario for Associating an Email with an Opportunity

This scenario gives one example of how you might use Oracle CRM On Demand Desktop to associate an email with an opportunity. You might use Oracle CRM On Demand Desktop differently, depending on your business model.

A sales representative opens an email from an external contact’s company. This email explains that the purchasing director at the external contact’s company has changed. The representative knows that this information is important for the Big Deal opportunity. The representative shares this email with the Oracle CRM On Demand server and associates it with the opportunity, making sure that the entire sales team who are working on the opportunity are aware that there is a new purchasing director. Oracle CRM On Demand Desktop synchronizes this information with the Oracle CRM On Demand server as an activity record that is associated with the opportunity, along with the original email as an activity attachment.

When the representative shares the email with the Oracle CRM On Demand server, the representative can choose the sales data with which the email is associated. To associate the external contact with the email, the representative clicks the Contacts button and selects the contact that is suggested in the list. To associate the Big Deal opportunity with the email, the representative types the name in the Opportunity control. If the external contact is associated with the Big Deal opportunity, then Oracle CRM On Demand Desktop presents the opportunity in the list of suggested opportunities when the representative clicks Opportunity. The representative chooses the record for the Big Deal opportunity, and then Oracle CRM On Demand Desktop links the email to the opportunity.

Scenario for Managing an Opportunity

This scenario gives one example of how you can use Oracle CRM On Demand Desktop to manage an opportunity. You might use Oracle CRM On Demand Desktop differently, depending on your business model.

After meeting with a contact at a strategic account, a sales representative learns of an upcoming Request For Proposal (RFP) that might help to improve the sales pipeline for the next quarter. To complete this work, the representative uses Oracle CRM On Demand Desktop to create a new opportunity in Microsoft Outlook.

To begin, the representative clicks Opportunity on the toolbar, which opens a new opportunity form. The representative enters details of the new opportunity, including the name, related account, lead quality, sales stage, close date, and so on. The representative knows two contacts at the account, and that these contacts decide whether or not to place an order. Therefore, the representative associates these contacts with the opportunity. The representative can choose one or more products and associate them with the opportunity. To indicate the projected value of the opportunity and to describe how that value is distributed across related products for the opportunity, the representative can assign expected revenue values for the opportunity.

When the representative saves these details in Microsoft Outlook and then synchronizes with the Oracle CRM On Demand server, Oracle CRM On Demand makes the details available to other users who have access to the account and contacts.

3 Overview of Oracle CRM On Demand Desktop Lite

Overview of Oracle CRM On Demand Desktop Lite

This chapter provides an overview of Oracle CRM On Demand Desktop Lite. It includes the following topics:

- *About Using Oracle CRM On Demand Desktop Lite*
- *About Oracle CRM On Demand Desktop Lite Privileges*
- *About Synchronizing Oracle CRM On Demand Desktop Lite*

About Using Oracle CRM On Demand Desktop Lite

Oracle CRM On Demand Desktop Lite is a client-side desktop application that is an extension to Microsoft Outlook. It allows the user to work with Oracle CRM On Demand in a native Microsoft Outlook environment. The data is accessible offline in Microsoft Outlook, and you can synchronize account, contact, appointment, and task records with the Oracle CRM On Demand server when Oracle CRM On Demand Desktop Lite is connected to the Internet.

Oracle CRM On Demand Desktop Lite is a lightweight version of Oracle CRM On Demand Desktop. Many of the features available in Oracle CRM On Demand Desktop are available in Oracle CRM On Demand Desktop Lite. However, there are some differences between Oracle CRM On Demand Desktop and Oracle CRM On Demand Desktop Lite. The following table lists the functionality of Oracle CRM On Demand Desktop and Oracle CRM On Demand Desktop Lite.

Feature	Desktop Lite	Desktop
Oracle CRM On Demand records synchronized	Contact and activity records are fully synchronized. Account records are partially synchronized. For more information on synchronizing Account records, see *Synchronizing Accounts*.	Contact, activity, account, opportunity, and lead. For more information on synchronization, see *How Oracle CRM On Demand Desktop Synchronizes Data*.
Synchronization depth	Child records are not synchronized. However, there is an exception for primary users. A child record is synchronized to maintain a record association. For more information, see *Synchronizing Primary Users*.	Child objects are synchronized. For more information on synchronization, see *How Oracle CRM On Demand Desktop Synchronizes Data*.
Synchronization limit	The synchronization limit is 3000 activity records, 2000 contact records, and 2000 account records.	The synchronization limit is 10,000 records. For more information on synchronization, see *How Oracle CRM On Demand Desktop Synchronizes Data*.
Activity associations	You can associate only the Primary Contact to an activity.	You can associate all contacts to an activity.

Feature	Desktop Lite	Desktop
Synchronization of email	You can synchronize your email.	You can synchronize your email.
Full synchronization frequency	Full synchronization frequency is once every four hours.	Full synchronization frequency is once every two hours.
Incremental synchronization frequency	Incremental synchronization is unlimited. For more information on synchronization, see *How Oracle CRM On Demand Desktop Synchronizes Data*.	Incremental synchronization is unlimited. For more information on synchronization, see *How Oracle CRM On Demand Desktop Synchronizes Data*.
Configuration	Configuration is not supported.	Configuration is supported.
Customization package is linked to Roles	The customization package is not linked to roles.	The customization package is linked to roles.
Duplication checking	Duplication checking is supported for account records.	Duplication checking is supported for account, activity, contact, lead, and opportunity records. Duplication checking is configurable.
Single Sign-On (SSO)	Single Sign-On is supported. For more information on SSO, see *Administering Single Sign-On*.	Single Sign-On is supported. For more information on SSO, see *Administering Single Sign-On*.
Online lookup	Online lookup is supported. For more information on online lookup, see *Configuring Online Lookup*.	Online lookup is supported. For more information on online lookup, see *Configuring Online Lookup*.
Record pinning	Record pinning is supported. For more information on record pinning, see *Configuring the Pin Mode for Records*.	Record pinning is supported. For more information on record pinning, see *Configuring the Pin Mode for Records*.

Installing Oracle CRM On Demand Desktop Lite

To install Oracle CRM On Demand Desktop Lite, you obtain and download either the 32 or 64-bit installer from the Oracle Software Delivery Cloud Web site. Procedures for installing Oracle CRM On Demand Desktop Lite are the same as Oracle CRM On Demand Desktop. For more information on installing Oracle CRM On Demand Desktop Lite, see *Process of Installing Oracle CRM On Demand Desktop*.

About Oracle CRM On Demand Desktop Lite Privileges

Your company administrator sets up and enables access to Oracle CRM On Demand Desktop Lite. To use Oracle CRM On Demand Desktop Lite, your role must include the Enable access for CRM Desktop Lite privilege. To administer Oracle CRM On Demand Desktop Lite, your role must include the Manage access for CRM Desktop Lite privilege.

About Synchronizing Oracle CRM On Demand Desktop Lite

You can use Oracle CRM On Demand Desktop Lite to synchronize account, contact, appointment, and task records between Oracle CRM On Demand and Microsoft Outlook. This allows you to update contact records, schedule appointments, and keep To-Do lists in either application and then synchronize the information. For more information on synchronization, see *How Oracle CRM On Demand Desktop Synchronizes Data with Oracle CRM On Demand*.

Synchronizing Contacts

New and updated contact records are synchronized between Microsoft Outlook and Oracle CRM On Demand. Contact activities are synchronized only when the contact is a primary contact for that activity.

You can associate the following information to contact records in Oracle CRM On Demand Desktop Lite:

- **Account information.** You can associate an existing account record (an account record that is already synchronized to Microsoft Outlook) to a contact record. Oracle CRM On Demand Desktop Lite only synchronizes the contact's primary account.
- **Email.** Email can be associated to a contact record. However, email attachments are not synchronized from Oracle CRM On Demand and Microsoft Outlook.
- **Team members.** Oracle CRM On Demand Desktop Lite only synchronizes the contact's primary team member.

Synchronizing Appointments and Tasks (Activities)

New, updated, or deleted activity records are synchronized between Microsoft Outlook and Oracle CRM On Demand. Oracle CRM On Demand Desktop Lite will only synchronize appointment and task records that are directly visible.

Note: Completed email activities are not synchronized by default.

You can associate the following information to activity records in Oracle CRM On Demand Desktop Lite:

- **Account information.** You can associate an existing account record (an account record that is already synchronized to Microsoft Outlook) to the corresponding activity record.
- **Contact information.** Oracle CRM On Demand Desktop Lite only synchronizes contact activities where the contact is the primary contact for that activity.
- **User information.** User is the only association object synchronized by Oracle CRM On Demand Desktop Lite in the same manner as Oracle CRM On Demand Desktop.

Synchronizing Accounts

New and updated account records are partially synchronized between Microsoft Outlook and Oracle CRM On Demand. The entire account record is not synchronized, only a few fields of the account record are synchronized. Oracle CRM On

Demand Desktop Lite synchronizes the following account fields: Account Name, Account Teams, Book Name, Location, and Description.

Email can be associated to account records and synchronized by Oracle CRM On Demand Desktop Lite. However, email attachments are not synchronized from Oracle CRM On Demand and Microsoft Outlook.

Synchronizing Primary Users

Child records are not synchronized between Microsoft Outlook and Oracle CRM On Demand. However, there are some child records that are synchronized because they are necessary to maintain record associations. For example, Oracle CRM On Demand Desktop Lite synchronizes the primary user of a contact or activity only if the contact is the primary contact for a contact or activity.

Synchronization Limits

The total number of records available to be synchronized is limited to 2000 contacts, 2000 accounts, and 3000 activities on the Oracle CRM On Demand server. If the limit is reached, then an error is returned. The error will not stop the overall synchronization process, just the synchronization of the impacted record. To avoid this error, you can use synchronization filters to reduce the number of Oracle CRM On Demand records that are available in Oracle CRM On Demand Desktop Lite. For more information on synchronization filters, see *Using Filters in Oracle CRM On Demand Desktop*.

The frequency of manual full synchronizations is monitored in Oracle CRM On Demand Desktop. If a user tries to perform a full synchronization within four hours of the last full synchronization, then a unique error code is generated.

Synchronization Deduplication and Conflict Resolution

Oracle CRM On Demand Desktop Lite supports deduplication and conflict resolution for accounts. For more information on deduplication and conflict resolution during synchronization, see *How Oracle CRM On Demand Desktop Handles a Conflict During Synchronization*.

4 How Oracle CRM On Demand Desktop Works

How Oracle CRM On Demand Desktop Works

This chapter describes how Oracle CRM On Demand Desktop works. It includes the following topics:

- *Extensions to the Microsoft Outlook User Interface*
- *Overview of Oracle CRM On Demand Desktop Architectural Components*
- *How Oracle CRM On Demand Desktop Uses Oracle CRM On Demand*
- *Metadata That Describes the Oracle CRM On Demand Desktop Application*

Extensions to the Microsoft Outlook User Interface

Oracle CRM On Demand Desktop extends Microsoft Outlook to store and display Oracle CRM On Demand sales data within Microsoft Outlook. The following are some examples of these extensions:

- Custom toolbar buttons
- Custom menu items
- Custom forms that display Oracle CRM On Demand data
- Custom controls that are embedded in Microsoft Outlook forms that display Oracle CRM On Demand data
- Personalization options dialog box
- Predefined Oracle CRM On Demand views in Microsoft Outlook folders, for example, the Oracle CRM On Demand Opportunities for Account view.

To allow the user to perform a variety of tasks, Microsoft Outlook uses these extensions. The following are some examples of work that the user can perform:

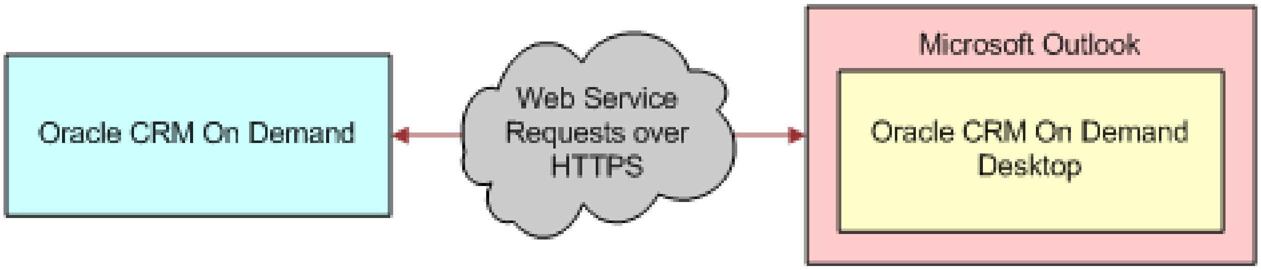
- Creating new Oracle CRM On Demand data in Microsoft Outlook
- Marking a Microsoft Outlook item to be shared with Oracle CRM On Demand data and associated sales data
- Viewing and editing sales data
- Starting a standard Microsoft Outlook action, such as sending an email or scheduling a meeting in the context of a sales item

Oracle CRM On Demand Desktop performs a variety of validations on data entry. The following are some examples of validation that Oracle CRM On Demand Desktop performs:

- Confirming that the data type is valid for a given field
- Making sure each required field includes information
- Making sure certain fields are disallowed, depending on the access rules for conditional data

Overview of Oracle CRM On Demand Desktop Architectural Components

The following figure illustrates the major architectural components that Oracle CRM On Demand Desktop uses.



Explanation of Callouts

Oracle CRM On Demand Desktop uses the following components:

- **Oracle CRM On Demand Desktop .** Oracle CRM On Demand Desktop is an extension of Microsoft Outlook, which uses a synchronization engine to determine the data changes between Microsoft Outlook and Oracle CRM On Demand. Oracle CRM On Demand data is saved locally in Microsoft Outlook, allowing for offline access of Oracle CRM On Demand data through Microsoft Outlook. For more information on using Oracle CRM On Demand Desktop with Microsoft Outlook, see *Extensions to the Microsoft Outlook User Interface*
- **Web Services.** Oracle CRM On Demand Desktop communicates with Oracle CRM On Demand, using Web services SOAP protocol. Oracle CRM On Demand Desktop submits requests and receives replies through Oracle CRM On Demand Web services to synchronize data between Microsoft Outlook and Oracle CRM On Demand. For more information on Oracle CRM On Demand Web services, see *About the Web Service API Used by Oracle CRM On Demand Desktop*.
- **Oracle CRM On Demand.** This is Oracle's customer relationship management (CRM) solution, which hosts CRM data and provides CRM functionality in a hosted Oracle CRM On Demand environment.

How Oracle CRM On Demand Desktop Synchronizes Data

To synchronize data between Microsoft Outlook and the Oracle CRM On Demand server, Oracle CRM On Demand Desktop uses a process that Oracle CRM On Demand Desktop controls. After Oracle CRM On Demand Desktop is installed, it initializes the Oracle CRM On Demand data that is available in Microsoft Outlook through the initial synchronization. An incremental synchronization synchronizes subsequent changes that occur in Microsoft Outlook or on the Oracle CRM On Demand server.

To make Oracle CRM On Demand data available in Microsoft Outlook, the Oracle CRM On Demand Desktop user must perform an initial synchronization with the Oracle CRM On Demand server. CRM On Demand Desktop Assistant guides the user through the setup of the Oracle CRM On Demand Desktop application in Microsoft Outlook. After you install Oracle CRM On Demand Desktop, it displays when the user starts Microsoft Outlook for the first time.

While using the CRM On Demand Desktop Assistant, the user can choose several installation preferences and then start the initial synchronization. Oracle CRM On Demand Desktop does the following:

1. Connects Microsoft Outlook to the Oracle CRM On Demand server and authenticates the user.
2. Checks the user's role to determine which customization package to retrieve from the Oracle CRM On Demand server.

   Customization packages are specific to a role in Oracle CRM On Demand. For more information customization packages, see *Files in the Customization Package*.
3. Performs a version check on the customization package, which verifies if a specific customization package already exists on the user's computer.

   For more information, see *Overview of Relationships Between Metadata Objects*.
4. Downloads and applies the configuration, as saved on the customization package.
5. Synchronizes the appropriate data. The connector configuration, synchronization mappings, visibility rules, default internal filters, and default user filters determine this synchronization.

   For more information, see *How Oracle CRM On Demand Desktop Synchronizes Data with Oracle CRM On Demand*.

About Web Service Usage During Synchronization

Microsoft Outlook and Oracle CRM On Demand use Web services to communicate with each other. Web service requests from Oracle CRM On Demand Desktop go through the normal Oracle CRM On Demand Web service infrastructure. Oracle CRM On Demand Desktop does the following:

- Sends Web service requests to verify the Oracle CRM On Demand Desktop version
- Queries Oracle CRM On Demand data
- Modifies Oracle CRM On Demand data by synchronizing changes made by the user in Microsoft Outlook to the Oracle CRM On Demand server

Web service responses are from Oracle CRM On Demand through the normal Oracle CRM On Demand Web service infrastructure. For more information, see *About the Web Service API Used by Oracle CRM On Demand Desktop*.

How Oracle CRM On Demand Desktop Displays Data in Microsoft Outlook

Oracle CRM On Demand Desktop stores Oracle CRM On Demand data in files that are native to Microsoft Outlook. It stores this data in one of the following files, depending on the data file that a user sets as the default mail delivery location in the Microsoft Outlook profile:

- A Microsoft Exchange folder file (.ost file) that is offline and cached
- A personal folders file (.pst file)

Microsoft Outlook data is created in the native Microsoft Outlook application. Examples include an appointment or calendar item.

Oracle CRM On Demand data can include the following items:

- Business data that the user creates in Oracle CRM On Demand Desktop
- Data that a user creates in Oracle CRM On Demand

- Data that resides in the Oracle CRM On Demand database on the Oracle CRM On Demand server

Examples of Oracle CRM On Demand data include an opportunity, account, or activity. Because Oracle CRM On Demand Desktop uses native Microsoft Outlook data files, Microsoft Outlook displays Oracle CRM On Demand data in native Microsoft Outlook UI elements, for example, lists and forms. Microsoft Outlook can display Oracle CRM On Demand data simultaneously with other Microsoft Outlook data while using the same UI element, such as a mailbox folder. The Oracle CRM On Demand Desktop user can specify a folder that displays Oracle CRM On Demand data and can also view Microsoft Outlook data in a Microsoft Outlook list view.

Oracle CRM On Demand Desktop displays Oracle CRM On Demand data in the following contexts:

- Microsoft Outlook forms, which are extensions to Microsoft Outlook calendar, contact, email, and tasks.
- Microsoft Outlook items, such as the details of an account or opportunity that is associated with a Microsoft Outlook appointment that is shared with Oracle CRM On Demand Desktop.

The data that Oracle CRM On Demand Desktop stores in cached Microsoft Exchange mode in OST file format or in personal folders in PST format is accessible to the user while disconnected. The user interacts with data that is accessible to Microsoft Outlook in the local or in the Exchange Server files. The user does not interact directly with data that resides on the Oracle CRM On Demand server.

How Oracle CRM On Demand Desktop Uses Oracle CRM On Demand

This topic describes some of the major components in Oracle CRM On Demand that Oracle CRM On Demand Desktop uses. It includes the following information:

- *About the Web Service API Used by Oracle CRM On Demand Desktop*
- *About Authentication and License Allotment*

About the Web Service API Used by Oracle CRM On Demand Desktop

Oracle CRM On Demand Desktop uses Oracle CRM On Demand Web services as well as Web service APIs specific to Oracle CRM On Demand Desktop.

> ⚠ **CAUTION:** Oracle CRM On Demand Desktop supports only stateless Web services. You must ensure that the Web Services R16 Compatibility Mode setting is not selected in the Oracle CRM On Demand Company Profile. For more information on the Oracle CRM On Demand Company Profile, see *Oracle CRM On Demand Online Help* .

A Client Sync Service Web service provides access to data and checks that the client versions are correct. It has one method,**ClientSyncService_Execute.** This method is a wrapper for the existing Oracle CRM On Demand Web Services v2.0 API. ClientSyncService_Execute calls the Web Services v2.0 QueryPage and Execute methods on Oracle CRM On Demand objects to synchronize data.

For more information about the Web Services v2.0 API, see *Oracle Web Services On Demand Guide* .

The Client Sync Service Web service is available through privilege checks and authorization only to Oracle CRM On Demand Desktop. A Web Services Description Language (WSDL) file is not published or available to customers. However, usage information for the Web service is visible in the Web Service Utilization page in Oracle CRM On Demand. Entries appear with a Web Service Name value of ClientSyncService and an Operation value of VersionCheck or Execute.

Oracle CRM On Demand Desktop always uses the Sales Rep value for the ViewMode argument in Web services calls. The Sales Rep view mode takes precedence over the Can Read All Records? setting in Oracle CRM On Demand. This setting means that a user can have the Can Read All Records? flag enabled for a record type, but the visibility to that record is blocked by the Sales Rep View mode unless the user is from the team that owns that record.

Visibility is also defined by the access settings defined for the role. For more information on roles and profiles, see *Oracle CRM On Demand Online Help* . For more information on the ViewMode argument, see *Oracle Web Services On Demand Guide* .

About Book-Based Visibility

In addition to visibility through record ownership and team membership settings, there is also a book visibility of records. For book visibility, you must set the ViewMode argument to Context in Web services API calls. For access to a specific book, the BookQueryPage method uses the BookId and BookName arguments. If the book has a parent book, then you must also set the ParentBookName argument through the same method. You can associate records with the book if you have the CanContainDataFlag setting enabled for that book. This setting corresponds to the Can Contain Data check box on a Book Detail page in the UI.

For more information on book functionality and configuration see *Oracle CRM On Demand Online Help* . For more information on Web services, see *Oracle Web Services On Demand Guide* .

About User Default Books

For access to a user default book, the UserQueryPage method uses the DefaultBookId and DefaultBookName arguments. You can also set default books for each record type. To obtain the default books for each record type, the UserDefaultBookPerType argument returns the default BookId, BookName, IncludeSubBooks, and AccessObjectName for each record type. If the default book for record type contains subbooks, then set the IncludeSubBooks flag to true.

About Authentication and License Allotment

To establish an authenticated session with Oracle CRM On Demand, users must have valid login credentials to Oracle CRM On Demand, with the following Web services privileges enabled:

- Enable Web Services Access
- Enable CRM Desktop Access

For more information on session management, see *Oracle Web Services On Demand Guide* .

The number of users that can be granted the Enable CRM Desktop privilege is limited to the number of Oracle CRM On Demand Desktop licenses that have been provisioned for the company.

You can view license allocation and usage in the Service Allotment Administration section of the Company Administration page of Oracle CRM On Demand. To adjust your license allotment for Oracle CRM On Demand Desktop, contact Oracle Customer Care.

How Oracle CRM On Demand Desktop Handles Allotment Errors

Multiple allotment errors can occur during the deployment phase and also during the synchronization process. The disrupted synchronization might be aborted or might restart when the allotment becomes available.

It is recommended that you use an accurate filter to limit the amount of data synchronized to the user. It is recommended that you purge unnecessary records and attachments to free space for Oracle CRM On Demand Desktop.

The following error is a license allotment error that the company administrator can receive when trying to assign a privilege to a role, or a role to a user:

You have exceeded the number of licenses for your company. Please contact your Oracle CRM On Demand Sales Representative to discuss licensing options.

This error occurs when you are trying to assign the Enable CRM Desktop privilege to too many users in your company. It is recommended that you either purchase additional licenses or that you move users to a new role to control the license assignment.

Checking the License Allotment in Oracle CRM On Demand

This topic describes how to check the license allotment in Oracle CRM On Demand.

To check the license allotment in Oracle CRM On Demand

- Go to the global Admin link, then Company Administration, and then Service Allotment Administration.

If you have to purchase license allotment for your company, then contact Oracle Customer Care.

Metadata That Describes the Oracle CRM On Demand Desktop Application

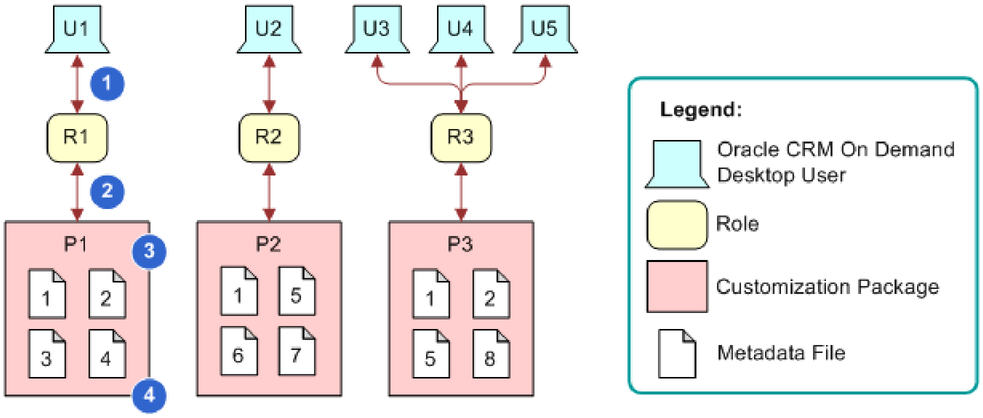
Oracle CRM On Demand Desktop is defined in metadata so that you can customize it to meet your business requirements. This topic describes the structure of the metadata that Oracle CRM On Demand Desktop uses and how you can modify the metadata to support a customization. It includes the following information:

- *Overview of Relationships Between Metadata Objects*
- *About the Customization Package*
- *About Metadata Files*

For more information on customizing an implementation of Oracle CRM On Demand Desktop, see *Customizing Oracle CRM On Demand Desktop*.

Overview of Relationships Between Metadata Objects

An Oracle CRM On Demand Desktop user is associated with a given Oracle CRM On Demand Desktop configuration through a role that a customization package references. When the package is activated and published, a user who is associated with the role can download the configuration that is defined in the package. This configuration is a collection of metadata files that Oracle CRM On Demand Desktop stores on the Oracle CRM On Demand server and downloads to Microsoft Outlook during synchronization. The following figure illustrates the relationships between users, roles, customization packages, and metadata files.



The following relationships exist between users, roles, customization packages, and metadata files:

1. **Oracle CRM On Demand Desktop user.** The user of an implementation of Oracle CRM On Demand Desktop.
2. **Role.** An Oracle CRM On Demand role, such as sales representative. It corresponds to the job role that the user performs.
3. **Customization package.** A collection of metadata files that Oracle CRM On Demand Desktop associates with a particular role. This package is delivered to the user in a ZIP file. For more information, see *About the Customization Package*.
4. **Metadata file.** A description of Oracle CRM On Demand Desktop that Oracle CRM On Demand Desktop sends to Microsoft Outlook as XML code or JavaScript files. Metadata files are contained in the customization package. For more information, see *About Metadata Files*.

How Oracle CRM On Demand Desktop Users Access Oracle CRM On Demand Data

Oracle CRM On Demand Desktop allows the user to access Oracle CRM On Demand data in Microsoft Outlook in the following ways:

- **Through roles.** In Oracle CRM On Demand, a customization package is associated with a role. This association determines which customization package Oracle CRM On Demand Desktop sends to the user. The metadata defines which objects Oracle CRM On Demand Desktop synchronizes with Microsoft Outlook.
- **Through synchronization filters.** The customization package includes metadata files that specify which data access control and which filters to apply when Oracle CRM On Demand Desktop synchronizes data with the Oracle CRM On Demand server. For example, the default configuration specifies that the user can synchronize accounts, contacts, leads and opportunities. These records are associated with the sales team to which the user is assigned.

Depending on your business requirements, you can define different customization packages and assign them to different users through the user’s roles. To meet individual user requirements, you can define different access control and synchronization filters for each customization package.

About the Customization Package

A *customization package* is a collection of XML metadata files and JavaScript files that Oracle CRM On Demand Desktop associates with a particular role. In Oracle CRM On Demand, the customization package is saved and used as a ZIP file containing the metadata files. If you want to modify the customization package, then you must unzip the customization package and make changes to the appropriate metadata files. When you are finished, you must rezip the entire customization package, and upload the modified customization package to the Oracle CRM On Demand server. For more information, see *Overview of Customizing Oracle CRM On Demand Desktop*. You can make the following customizations:

- Add or remove fields that Oracle CRM On Demand Desktop synchronizes.
- Change the layout of a custom form in the client.
- Change a control that Oracle CRM On Demand Desktop uses with Microsoft Outlook.
- Change a security rule.

The customization package describes the following information:

- The extensions to the Oracle CRM On Demand Desktop UI, which includes Microsoft Outlook views, forms, lookup controls, and toolbars
- Translated text strings that Oracle CRM On Demand Desktop uses to create prompts and labels in Microsoft Outlook
- Data validation and security logic
- Descriptions of synchronization preset filters and view modes that Oracle CRM On Demand Desktop uses during synchronization
- Criteria that Oracle CRM On Demand Desktop uses to detect duplicate objects that occur in Oracle CRM On Demand Desktop
- Business logic that JavaScript provides
- Data mapping between Oracle CRM On Demand fields and Microsoft Outlook fields

If you change the data model in the customization package, then Oracle CRM On Demand Desktop performs a complete resynchronization. For more information on customizing the customization package, see *Customizing Oracle CRM On Demand Desktop*.

Guidelines for Assigning a Role to a Customization Package

When you develop a customization package you must follow these guidelines:

- **You can associate only one customization package with a role.** A user can be assigned to only one role. However, a role can have many users.
- **You can assign the same customization package to multiple roles.** However when you do this assignment you must create a separate package record for each role. You manage roles in the Company Administration section of Oracle CRM On Demand.
- **Only one customization package can be published for a specific role at any time.**

About Metadata Files

A metadata file is an XML or JavaScript file that describes the Oracle CRM On Demand Desktop application. Oracle CRM On Demand Desktop uses these files in the following ways:

- **XML files.** Describes the default synchronization objects, synchronization mapping, custom views and forms in Microsoft Outlook, and so on.
- **JavaScript files.** Describes the business logic that Oracle CRM On Demand Desktop uses for data validation, custom actions that Oracle CRM On Demand Desktop provides access to in toolbars, and other custom processing that Oracle CRM On Demand Desktop performs in Microsoft Outlook.

The following items describe how Oracle CRM On Demand Desktop uses metadata files with a customization package:

- A customization package contains a collection of metadata files. These files describe the entire Oracle CRM On Demand Desktop application that you use with Microsoft Outlook.
- A customization package consists of a set of metadata files.
- Oracle CRM On Demand Desktop requires that a customization package include a minimum number of files. These files are described in *Files in the Customization Package*.
- You can use a single metadata file with more than one customization package. These metadata files can be part of another customization package. The figure in the topic *Overview of Relationships Between Metadata Objects* illustrates this relationship where the same metadata file occurs in different packages, for example, packages 1 and 3 include metadata file 2.

Oracle CRM On Demand Desktop associates a customization package with a single Oracle CRM On Demand role. It associates a user with the role as a way to acquire access to the customization package and the Oracle CRM On Demand Desktop configuration that you use with Microsoft Outlook. For more information, see *Files in the Customization Package*.

How Oracle CRM On Demand Desktop Reuses, Modifies, and Updates Metadata Files

Oracle CRM On Demand Desktop stores the metadata files in a customization package. This customization package is stored on Oracle CRM On Demand as a ZIP file. In Oracle CRM On Demand, metadata files are not tracked or stored individually.

Although you can use an existing role and associate it with a customization package, it is recommended that you create a new role. For example, the Oracle CRM On Demand sales representative role is different and separate from the Oracle CRM On Demand sales and marketing manager role. Creating a new role provides you with more control over which users Oracle CRM On Demand Desktop associates with which customization package. For more information, see *Creating and Publishing the Customization Package*.

5 How Oracle CRM On Demand Desktop Handles Oracle CRM On Demand Data

How Oracle CRM On Demand Desktop Handles Oracle CRM On Demand Data

This chapter describes how Oracle CRM On Demand Desktop handles certain types of Oracle CRM On Demand data. It includes the following topics:

- *How Oracle CRM On Demand Desktop Handles an Activity*
- *How Oracle CRM On Demand Desktop Handles a Shared Activity*
- *How Oracle CRM On Demand Desktop Handles Microsoft Outlook Calendar*
- *How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Task*
- *How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Email Message*
- *How Oracle CRM On Demand Desktop Handles Items If the User Removes Oracle CRM On Demand Desktop*
- *How a User Can Link an Oracle CRM On Demand Record to a Microsoft Outlook Record*

How Oracle CRM On Demand Desktop Handles an Activity

This topic describes how Oracle CRM On Demand Desktop handles an activity. It includes the following information:

- *Overview of How Oracle CRM On Demand Desktop Handles an Activity*
- *How an Activity Is Created or Modified*
- *How Oracle CRM On Demand Desktop Processes an Activity*
- *How Oracle CRM On Demand Desktop Resolves Participants and Email Recipients of an Activity*
- *How Oracle CRM On Demand Desktop Displays an Activity in Microsoft Outlook*
- *How Oracle CRM On Demand Desktop Sets the Owner of an Activity*

Overview of How Oracle CRM On Demand Desktop Handles an Activity

In Oracle CRM On Demand, an activity is a work item that the user must track or display as an interaction. The following items are examples of activities:

- A task item

- An email sent to a contact
- An appointment that includes a contact

Oracle CRM On Demand can display an activity in the Home tab or the calendar. The Activity field of the Activities list determines where in Microsoft Outlook an activity displays. The following values are included:

- Appointment
- Task

The Type field specifies the type of activity. It can contain a wide range of possible values, including the following:

- Appointment
- Field Repair
- Email
- Research

An Oracle CRM On Demand activity does not map to a single native object in Microsoft Outlook. Instead, Oracle CRM On Demand Desktop synchronizes an activity from the Oracle CRM On Demand server to Oracle CRM On Demand Desktop as a custom activity record rather than as a Microsoft Outlook task or calendar item. After synchronization, Oracle CRM On Demand Desktop does the following:

- Creates a Microsoft Outlook calendar item that matches the appointment activity from Oracle CRM On Demand
- Creates a Microsoft Outlook task item that matches the task activity from Oracle CRM On Demand

Because Microsoft Outlook does not synchronize directly between native Microsoft Outlook items and records on the Oracle CRM On Demand server, Oracle CRM On Demand Desktop uses the Oracle CRM On Demand activity as an intermediary between a native Microsoft Outlook item that resides in the user mailbox and an Oracle CRM On Demand activity that resides on the Oracle CRM On Demand server. If the user creates a shared Microsoft Outlook appointment, email, or task, then Oracle CRM On Demand Desktop creates another item in Microsoft Outlook that represents the Oracle CRM On Demand activity record in addition to the native Microsoft Outlook item that is shared.

You also have the option to determine how Oracle CRM On Demand Desktop shares newly created Microsoft Outlook items such as appointments, contacts and tasks with Oracle CRM On Demand. This is controlled by the SharedByDefault:NewItems Windows registry key. For more information on Windows registry keys, see *Setting Behavioral Limits for Oracle CRM On Demand Desktop*. The following procedure describes how to access this option in Oracle CRM On Demand Desktop.

Controlling How Oracle CRM On Demand Desktop Shares Newly Created Microsoft Outlook Items

The following procedure describes how to control how Oracle CRM On Demand Desktop shares newly created Microsoft Outlook items, such as appointments, contacts and tasks.

To control how Oracle CRM On Demand Desktop shares newly created Microsoft Outlook items

1. Right-click the Oracle CRM On Demand Desktop icon in the system tray, and click Options.
2. Click Advanced, and select Appointments, Contacts, Tasks in the Oracle CRM On Demand Desktop - Options dialog box.

   Microsoft Outlook will always share new appointments, contacts, and tasks with Oracle CRM On Demand.

How an Activity Is Created or Modified

The user can use one of the following ways to perform read, update, create, and delete operations:

- Use the Add Appt or Add Task buttons.
- Use a form for an item in Microsoft Outlook that includes a relationship with an activity, such as all the activities for an account. This form allows the user to link the activity with an Oracle CRM On Demand record in Microsoft Outlook, such as an account, opportunity, lead, or contact, and to display the link to the corresponding activity.

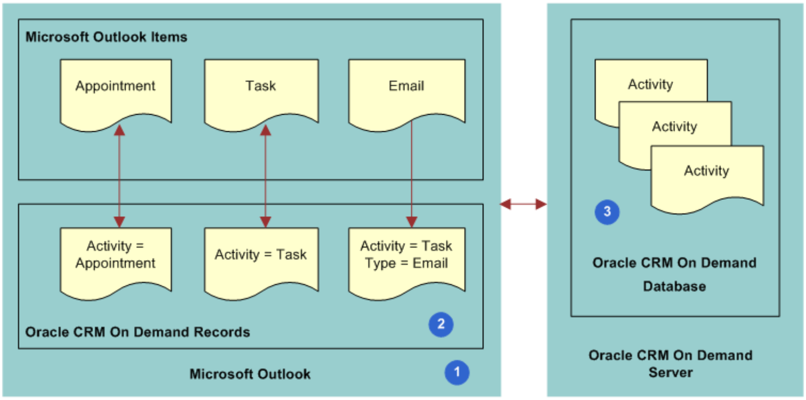
Oracle CRM On Demand Desktop can create an activity for an item in Microsoft Outlook, such as an appointment in the calendar, a task, or an email.

How Oracle CRM On Demand Desktop Processes an Activity

To process an activity, Oracle CRM On Demand Desktop uses the following types of objects:

- A native Microsoft Outlook item, such as an appointment in the calendar, an email, or a task
- An Oracle CRM On Demand activity record in Microsoft Outlook that Oracle CRM On Demand Desktop synchronizes from the Oracle CRM On Demand server
- An Oracle CRM On Demand activity record on the Oracle CRM On Demand server

The following figure illustrates the relationships between Microsoft Outlook items in Microsoft Outlook, Oracle CRM On Demand records in Microsoft Outlook, and Oracle CRM On Demand records on the Oracle CRM On Demand server. Multiple activity types map to the Activity value of the Activities Only list, for example, demos, and so on. For brevity, the following figure does not include these types.



Explanation of Callouts

To process an activity, the following work occurs:

1. An activity is created in Microsoft Outlook.

   For more information, see *How an Activity Is Created or Modified*.
2. Oracle CRM On Demand Desktop adds a record as an Oracle CRM On Demand activity in Microsoft Outlook. If the user marks as shared a Microsoft Outlook appointment, email, or task, and then saves and closes this item, then Oracle CRM On Demand Desktop immediately creates the Oracle CRM On Demand activity in Microsoft Outlook.

3. During synchronization, Oracle CRM On Demand Desktop maps the Oracle CRM On Demand activity in Microsoft Outlook as a one-to-one relationship with the corresponding activity in the Oracle CRM On Demand database on the Oracle CRM On Demand server. If the user shares a new Microsoft Outlook item or creates an activity in Microsoft Outlook, then Oracle CRM On Demand Desktop uploads this activity during synchronization to the Oracle CRM On Demand server and inserts it in the Oracle CRM On Demand database.

   For more information, see *How Oracle CRM On Demand Desktop Creates a Corresponding Native Microsoft Outlook Item*.

How Oracle CRM On Demand Desktop Creates a Corresponding Native Microsoft Outlook Item

When synchronizing data, if the user has the privileges to view the activity, and if the activity meets the requirements that the filter settings for the user defines, then Oracle CRM On Demand Desktop downloads any new activities to Microsoft Outlook that reside on the Oracle CRM On Demand server.

The following table describes how Oracle CRM On Demand Desktop creates a corresponding native Microsoft Outlook item depending on the settings of the Activity field.

Activity Value for the Oracle CRM On Demand Activity	Description
Appointment	Oracle CRM On Demand Desktop creates a calendar item as an appointment only. A start date is required by the Oracle CRM On Demand server to create a calendar item in Microsoft Outlook Calendar. The calendar item is linked to this activity.
Task	Oracle CRM On Demand Desktop creates a native Microsoft Outlook task that is linked with this activity.

How Oracle CRM On Demand Desktop Resolves Participants and Email Recipients of an Activity

This topic describes how Oracle CRM On Demand Desktop resolves participant lists and email recipients in Microsoft Outlook calendar.

How Oracle CRM On Demand Desktop Resolves Attendees of a Meeting

Oracle CRM On Demand Desktop performs the following actions:

- If the meeting organizer adds an email address in the To line, then Oracle CRM On Demand Desktop creates an association with a user or contact.
- If the meeting organizer uses an MVG (Multi Value Group) in the activity form to add an association, then Oracle CRM On Demand Desktop does not update the To line. An MVG is a list of records of the same type that is associated with a parent record. For example, a user can link multiple contacts to a task activity (parent record). This link allows the user to associate the Oracle CRM On Demand activity with a contact but not invite the contact to the meeting.

How Oracle CRM On Demand Desktop Resolves a Task Owner and Assignees of a Task

Oracle CRM On Demand Desktop does not add the email addresses to the To field if the associations to users or contacts are made through the User or the Contact MVG dialog box for the activity that is linked to a shared task. Creating an association with a user or a contact does not assign the task to this user or contact.

How Oracle CRM On Demand Desktop Resolves Recipients of an Email

Oracle CRM On Demand Desktop performs the following actions:

- If the user shares an email that a user manually sent to or received from a contact, then Oracle CRM On Demand Desktop does the following:
  - Suggests associations for resolved contacts for the email activity.
  - Suggests a list of accounts and opportunities that are related to the resolved contacts.
- If the user chooses an account or opportunity, then Oracle CRM On Demand Desktop associates the account or opportunity with the activity that the user creates from the email.
- If the user sends a shared email, then Oracle CRM On Demand Desktop does the same processing except it does not resolve the sender as a contact.
- If the user manually associates the shared email with an Oracle CRM On Demand record before the user sends the email, then the automail processing feature does the following:
  - Preserves the associations that the user makes
  - Updates the email activity with contact associations that Oracle CRM On Demand Desktop resolves from the email addresses of the recipients.
- If, to associate a contact, the user uses the Contact MVG dialog box for an activity that is linked to a shared email, then Oracle CRM On Demand Desktop does not add email addresses to the To field. An association that is created to a contact does not update the recipients list for the email message.

How Oracle CRM On Demand Desktop Displays an Activity in Microsoft Outlook

Oracle CRM On Demand Desktop displays data for an Oracle CRM On Demand activity in Microsoft Outlook in the following ways:

- For a shared appointment, email, or task, Oracle CRM On Demand Desktop displays details of the related activity in the native Microsoft Outlook form. For example, the native Microsoft Outlook appointment form displays the following information:
  - The subject of the Oracle CRM On Demand activity in the Subject field of the native Microsoft Outlook Calendar item.
  - The account that is associated with this Oracle CRM On Demand activity in the Account field in the extended area of the form.

    For example, to review and change the account, opportunity, contacts, leads and users for the Oracle CRM On Demand activity that is related to the shared item, the user can use the native Microsoft Outlook form.
- As a list of Oracle CRM On Demand activity records that are related to a parent sales record, for example, a list of activities that are related to an account or opportunity.

By default, Oracle CRM On Demand Desktop displays a folder for activities in the Microsoft Outlook navigation pane.

How Oracle CRM On Demand Desktop Sets the Owner of an Activity

This topic describes how Oracle CRM On Demand Desktop sets the owner of an activity, for an appointment or a task.

How Oracle CRM On Demand Desktop Sets the Owner of an Appointment

Oracle CRM On Demand Desktop sets the owner of an appointment according to the record ownership mode of the activity type, as follows:

The Record Ownership Mode is Set to Owner Mode

In this case, Oracle CRM On Demand Desktop sets the OwnerId field of an appointment according to the following priority:

1. Oracle CRM On Demand Desktop resolves the email address of the native Microsoft Outlook appointment to an Oracle CRM On Demand user. If Oracle CRM On Demand Desktop finds a user record that contains this address, then it sets the meeting organizer of the Microsoft Outlook Calendar event as the owner.
2. If Oracle CRM On Demand Desktop does not find a user that contains this address, then it compares this address with addresses from email accounts in the Microsoft Outlook profile. If it finds a match, then it returns the user from the user object. This situation can occur if the email address that is set for the current user is not the same as the account address in the native Microsoft Outlook record for this user.
3. If Oracle CRM On Demand Desktop does not find a match among the email accounts in the Microsoft Outlook profile, then no user is found. In this situation, Oracle CRM On Demand Desktop sets the owner to the value in the Current User. For more information, see *How Oracle CRM On Demand Assigns the Meeting Organizer*.

The Record Ownership Mode is Set to Book Mode

In this case, Oracle CRM On Demand Desktop sets the BookId field of an appointment according to the following priority:

1. Oracle CRM On Demand Desktop extracts the default book for the activity type from the picklist containing the default books for each record type and sets this selected book as the appointment's book.
2. If no default book for the activity type is specified in the picklist, then Oracle CRM On Demand Desktop extracts the default book for the user from the corresponding picklist and sets this selected book as the appointment's book.
3. If no default book for the user is specified in the picklist, then the user is prompted to select the book for the appointment.

The Record Ownership Mode is Set to Mixed mode

In this case, Oracle CRM On Demand Desktop sets either the BookId or OwnerId field of an appointment according to the following priority:

1. Oracle CRM On Demand Desktop extracts the default book for an activity type from the picklist containing the default books for each record types and sets this selected book as the appointment's book.
2. If no default book for the activity type is specified in the picklist, then Oracle CRM On Demand Desktop attempts to populate the OwnerId field, using the same algorithm as *The Record Ownership Mode is Set to Owner Mode*.

How Oracle CRM On Demand Assigns the Meeting Organizer

An Oracle CRM On Demand user is a user who is registered to use Oracle CRM On Demand Desktop or Oracle CRM On Demand. The meeting organizer is the user who creates the meeting. If a user creates a meeting, then Oracle CRM On Demand does the following:

- If the meeting organizer is an Oracle CRM On Demand user, then Oracle CRM On Demand sets the value in the Owner field of the activity to the following value:

  Meeting Organizer

- If the meeting organizer is not an Oracle CRM On Demand user, then Oracle CRM On Demand sets the value in the Owner field of the activity to Current User.

How Oracle CRM On Demand Assigns the Meeting Organizer If The Organizer Is Not an Oracle CRM On Demand User

Oracle CRM On Demand requires the following:

- Every activity must include an owner, if the activity uses owner-only record ownership mode.
- An Oracle CRM On Demand user record must exist for this owner.

Assume an Oracle CRM On Demand user creates an appointment in Microsoft Outlook. In this situation, a user record exists for this user, so Oracle CRM On Demand Desktop sets this user as the owner and then synchronizes this appointment with the Oracle CRM On Demand server.

A user record does not exist in the following situations:

- Assume user A in your organization is not an Oracle CRM On Demand user. This user creates a meeting and then invites another user in your organization (who is an Oracle CRM On Demand user) to this meeting. An Oracle CRM On Demand user record does not exist for User A, and this user cannot own an Oracle CRM On Demand record.
- A contact who is external to your company creates a meeting. An Oracle CRM On Demand contact cannot own a meeting.

To create the meeting in this situation, Oracle CRM On Demand must first determine the owner for this activity. To avoid duplication errors and access conflicts between users for this meeting, Oracle CRM On Demand does the following:

1. Creates a meeting.
2. Assigns the Current User as the owner of this meeting.

How Oracle CRM On Demand Desktop Sets the Owner of a Task

This topic describes how Oracle CRM On Demand Desktop sets the owner for a task for different ownership modes.

How Oracle CRM On Demand Desktop Prefills with a Default Book

Throughout this topic, the phrase prefills with default book refers to the following logic performed by Oracle CRM On Demand Desktop:

1. Oracle CRM On Demand Desktop extracts the book selected by the user for an Activity type in the Record set tab on the Control Panel.
2. If the book selected in the Record Set tab cannot contain data, than Oracle CRM On Demand Desktop extracts the default book for the Activity type specified on the Oracle CRM On Demand server, and sets this selected book as the appointment's book.
3. If no default book for Activity type is specified on the Oracle CRM On Demand server, then Oracle CRM On Demand Desktop extracts the default book for the user specified on the Oracle CRM On Demand server and sets this selected book as the appointment's book.
4. If no default book for the user is specified on the Oracle CRM On Demand server, than the user is prompted to select the book for the appointment.

How Oracle CRM On Demand Desktop Sets the Owner of a Task for Mixed Mode

If mixed mode is selected, then Oracle CRM On Demand Desktop checks the settings specified on Record Set tab on Control Panel and uses the following logic:

- The logic pertaining to Team and Owner mode if you have selected the `All records where I am owner or on the team` setting.
- The logic pertaining to Book mode if you have selected the `All records in a book` or the `All records in my default book` settings.

How Oracle CRM On Demand Desktop Sets the Owner of a Task for Owner and Team Ownership Mode or Book Mode

The following list describes how Oracle CRM On Demand Desktop sets the owner of a task for Owner and Team ownership mode or for Book ownership mode:

- If a user creates a shared task that is shared only with the user and:

    - If the Owner and Team ownership mode is selected, then Oracle CRM On Demand Desktop resolves the user as the owner of the Oracle CRM On Demand activity.
    - If Book ownership mode is selected, then Oracle CRM On Demand Desktop associates the task creator with the activity team but does not make the creator a primary team member, and so there is no owner. Oracle CRM On Demand Desktop also prefills the activity with the default book.

- If a user creates a task that is shared and delegated, and if the user keeps a copy of the tasks in the user mailbox, then Oracle CRM On Demand Desktop creates an activity and sets the owner according to the following rules:

    - If the user delegates the task only to another user, then Oracle CRM On Demand Desktop does the following:

        - Creates an activity in Microsoft Outlook for the user.

          Note: In this case, Oracle CRM On Demand Desktop specifies this user as an association with the user's team.

        - If the Owner and Team ownership mode is selected, then Oracle CRM On Demand Desktop sets the first user in the To line as the owner.

          If Book ownership mode is selected, then Oracle CRM On Demand Desktop associates the creator and the user with the activity team but does not make either of them primary team member, and so there is no owner. Oracle CRM On Demand Desktop also prefills the activity with the default book.
    - If the user delegates the task of a shared contact to a mixture of shared, unshared, or native Microsoft Outlook contacts, then Oracle CRM On Demand Desktop does the following:

        - Creates the activity.
        - If the Owner and Team ownership mode is selected, then Oracle CRM On Demand Desktop sets the user as the owner.

          If Book ownership mode is selected, then Oracle CRM On Demand Desktop associates the creator with the activity team but does not make the creator a primary team member, and so there is no owner. Oracle CRM On Demand Desktop also prefills the activity with the default book.
        - Associates all shared contacts that it resolved from email addresses in the task To line. It makes these associations in the Contacts list.
    - If the user delegates the task to shared contacts and users, then Oracle CRM On Demand Desktop does the following:

        - If the Owner and Team ownership mode is selected, then Oracle CRM On Demand Desktop sets the first user in the To line as the owner.

          If Book ownership mode is selected, then Oracle CRM On Demand Desktop associates all users in To line with the activity team but does not make any of them primary team member, and so there is no owner. Oracle CRM On Demand Desktop also prefills the activity with the default book.
        - Associates the creator with the activity team.
        - Associates contacts with the Contacts list.
        - If the Owner and Team ownership mode is selected, then Oracle CRM On Demand Desktop does not create associations with other users.

    If the Owner and Team ownership mode is selected, then this configuration helps to avoid having two similar activities for the same user:

    - For each assigned user who accepts the task, Oracle CRM On Demand Desktop creates an activity with this user, sets the owner, and associates the task creator to the user's team. Oracle CRM On Demand Desktop does not create any other associations.
    - The activity that Oracle CRM On Demand Desktop creates in Microsoft Outlook for the first user in the task To line is the same as the activity that it creates in Microsoft Outlook for the task creator.
  - If the user delegates the task to an external contact, and:
    - If the Owner and Team ownership mode is selected, then Oracle CRM On Demand Desktop creates an activity and sets the creator as the owner.
    - If Book ownership mode is selected, then Oracle CRM On Demand Desktop associates the creator with the activity team but does not make the creator a primary team member, and so there is no owner. Oracle CRM On Demand Desktop also prefills the activity with the default book.
  - If a user receives and shares a task, and:
    - If the Owner and Team ownership mode is selected, then Oracle CRM On Demand Desktop creates the activity, sets the user who received the task as the owner and adds the user who sent the task to the user's team as a nonprimary member.
    - If Book ownership mode is selected, then Oracle CRM On Demand Desktop associates the creator and the user who sent the task with the activity team but does not make either of them primary team members, and so there is no owner. Oracle CRM On Demand Desktop also prefills the activity with the default book.
  - If a user changes the book on a task to a book that is different from the one selected on the Record Set tab on the Control Panel, and if it is a book that is synchronized between Oracle CRM On Demand Desktop and Oracle CRM On Demand, then Oracle CRM On Demand Desktop deletes the activity corresponding to the Microsoft Outlook task, but does not delete the task itself.

Ownership Modes of Oracle CRM On Demand Desktop

You can apply the following ownership modes to the record types with owner and book-based visibility:

- **User Only mode.** In User only mode, the owner is required for the record type. In this case, Oracle CRM On Demand Desktop disregards the Record Set settings from the Synchronization Control Panel. Other users can be added as team members.
- **Book Only mode.** In Book only mode, the book is required for the record type. In Oracle CRM On Demand Desktop, records can be associated only with those books that can contain data that is configured in Oracle CRM On Demand.
- **Mixed mode.** In Mixed mode, neither the owner nor the book is specified for the record type. You can set the owner-based or book-based visibility options on the Record Set tab of the Synchronization Control panel.

How Oracle CRM On Demand Desktop Handles a Shared Activity

Microsoft Outlook supports the concept where more than one user can be associated with the same meeting or task. In this situation, Oracle CRM On Demand Desktop prevents the creation of duplicate Oracle CRM On Demand activities. It associates only one Oracle CRM On Demand activity with a single Microsoft Outlook item when more than one user synchronizes that activity. The first user who synchronizes the Microsoft Outlook item creates the Oracle CRM On Demand activity on the Oracle CRM On Demand server. Oracle CRM On Demand Desktop links Microsoft Outlook items with the Oracle CRM On Demand activity for any subsequent user who synchronizes activities.

To prevent duplicate records, Oracle CRM On Demand Desktop does the following:

1. Uses the unique identifier for the meeting and task that Microsoft Outlook provides.
2. Enters information in the iCRMId field on the Oracle CRM On Demand activity with the unique identifier from Step 1.
3. During synchronization, Oracle CRM On Demand Desktop validates that no other Oracle CRM On Demand activity includes this same unique identifier.
4. If Oracle CRM On Demand Desktop finds that there is no other activity, then it creates a new activity.
5. If Oracle CRM On Demand Desktop finds that there is another activity, then it downloads the existing activity with the same unique identifier, and then links it with the Microsoft Outlook item.

For more information, see *How Oracle CRM On Demand Desktop Prevents Duplicate Records*.

How the Origin of an Activity Affects Handling

This topic describes how the origin of an activity affects how the activity is managed in Oracle CRM On Demand and Microsoft Outlook.

Handling an Item from Oracle CRM On Demand

In this situation, Oracle CRM On Demand creates the activity on the Oracle CRM On Demand server. When Oracle CRM On Demand downloads this activity from the Oracle CRM On Demand server, it creates a native Microsoft Outlook item if the current user is the owner of the task, or if the current user is in the meeting participant list. Oracle CRM On Demand creates the activity as a simple appointment. No additional handling occurs in Microsoft Outlook.

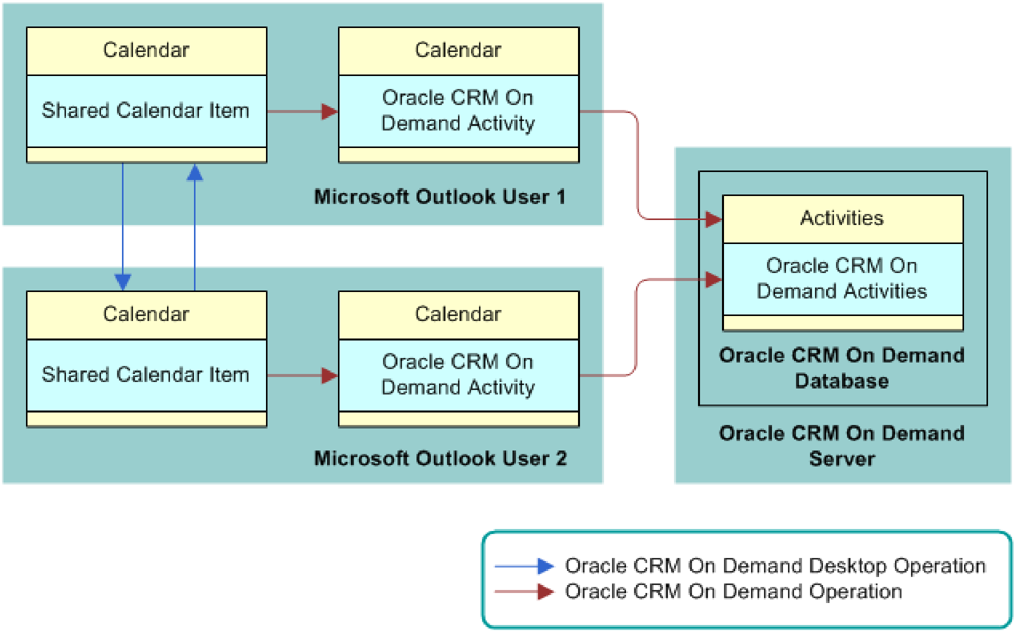
Handling an Item from Microsoft Outlook

In this situation, the activity is created in Microsoft Outlook, uploaded to the Oracle CRM On Demand server during synchronization and then downloaded to another user during an incremental synchronization. When Oracle CRM On Demand Desktop downloads this activity from the server, it does not add an item in the Microsoft Outlook calendar. Instead, Oracle CRM On Demand Desktop expects Microsoft Outlook to create the necessary item in the user mailbox. To create the item, Microsoft Outlook executes the native process it uses to send a meeting invitation or to assign a task.

If Oracle CRM On Demand Desktop does this work before it synchronizes the Oracle CRM On Demand activity with the Oracle CRM On Demand server, then Oracle CRM On Demand Desktop links the Oracle CRM On Demand activity with the meeting or task and then displays the item in shared mode. In shared mode, the share bar is active and any related details of the Oracle CRM On Demand activity display in the extended area of the native Microsoft Outlook form. If the user shares the item, then the item might not include all the details that the meeting organizer or task owner specified. These details arrive only after Oracle CRM On Demand Desktop synchronizes the Oracle CRM On Demand activity from the Oracle CRM On Demand server.

Example of How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Meeting with Multiple Attendees

The following figure illustrates how user 1, who is a meeting organizer, creates a native calendar item in Microsoft Outlook and shares it. User 2, the invitee, accepts the invitation and also shares it.


Calendar
Shared Calendar Item
Calendar
Oracle CRM On Demand Activity
Microsoft Outlook User 1
Calendar
Shared Calendar Item
Calendar
Oracle CRM On Demand Activity
Microsoft Outlook User 2
Activities
Oracle CRM On Demand Activities
Oracle CRM On Demand Database
Oracle CRM On Demand Server
Oracle CRM On Demand Desktop Operation
Oracle CRM On Demand Operation

The following process is performed:

1. User 1, the meeting organizer, creates a shared meeting with user 2, a meeting attendee who is a user.
2. User 1 saves the meeting and sends an invitation to user 2. The ConversationId is the same for the organizer and other meeting attendees.

   A ConversationId is the name of the field of Microsoft Outlook PIM items (Email, Appointment and Task) and is used to link a PIM item to an activity. When a PIM item is shared and a new related activity is created, the ConversationId field value is copied to the activity's iCRMId field so that both identifiers are the same. The activity's iCRMId is then synchronized with Oracle CRM On Demand as a part of the activity's field set. This synchronization makes it possible to link activities that are synchronized from Oracle CRM On Demand to PIM items.
3. User 1 synchronizes, and then Oracle CRM On Demand Desktop creates the activity on the Oracle CRM On Demand server.
4. User 2 receives and accepts the invitation.
5. User 2 shares the meeting, saves it, and then synchronizes. Oracle CRM On Demand Desktop determines user 1 already synchronized this Microsoft Outlook meeting with the Oracle CRM On Demand server because these items use the same ConversationId and include the same meeting organizer. In this situation, Oracle CRM On Demand Desktop identifies a duplicate during synchronization.
6. Oracle CRM On Demand Desktop detects that the activities are the same, and it then proceeds to perform deduplication conflict resolution without displaying an error message.

Example of How Oracle CRM On Demand Desktop Handles a Shared Microsoft Outlook Appointment That Is Declined

This topic describes an example of how Oracle CRM On Demand Desktop handles a Microsoft Outlook appointment that the meeting organizer shares and that the meeting attendee declines. In this situation, the user receives an invitation from another user through Microsoft Outlook and then declines the invitation:

1. User 1, the meeting organizer, creates an appointment, shares it, and then sends an invitation to user 2, to the meeting.
2. User 1 synchronizes. Oracle CRM On Demand Desktop uploads the activity to the Oracle CRM On Demand server.

3. User 2 receives the meeting invitation. Oracle CRM On Demand Desktop creates a shared meeting in Microsoft Outlook for User 2. Oracle CRM On Demand Desktop shares the meeting because the preference for User 2 is to create new native Microsoft Outlook items as shared. Oracle CRM On Demand Desktop creates the Oracle CRM On Demand activity with User 2 in the user team.
4. User 2 declines the meeting. Following the meeting rejection, a send notification is sent to the organizer.
5. Oracle CRM On Demand Desktop deletes the appointment from user 2's calendar.
6. If user 2 declines the shared meeting, and if the activity is not synchronized with the Oracle CRM On Demand server, then Oracle CRM On Demand Desktop deletes the activity in Microsoft Outlook for user 2. The same situation applies for any meeting participant who unshares or deletes the shared meeting request.
7. User 1 receives the decline notification, Oracle CRM On Demand Desktop updates Microsoft Outlook and removes user 2 from the user team.
8. User 1 synchronizes. Oracle CRM On Demand Desktop synchronizes the changes with the Oracle CRM On Demand server.
9. User 2 synchronizes.

How Oracle CRM On Demand Desktop Handles Microsoft Outlook Calendar

This topic describes how Oracle CRM On Demand Desktop handles data for a native Microsoft Outlook calendar item. It includes the following information:

- *How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Calendar Item That the User Saves, Changes, or Deletes*
- *How Oracle CRM On Demand Desktop Handles an Oracle CRM On Demand Activity That the User Saves, Changes, or Deletes*
- *How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Email Message*
- *How Oracle CRM On Demand Desktop Handles Items If the User Removes Oracle CRM On Demand Desktop*

How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Calendar Item That the User Saves, Changes, or Deletes

The following actions occur if the user saves, changes, or deletes an item in the Microsoft Outlook calendar:

- If the user saves a new Microsoft Outlook calendar item that is shared, then Oracle CRM On Demand Desktop creates a new Oracle CRM On Demand activity in Microsoft Outlook.
- If the user changes an appointment in the Microsoft Outlook calendar, and if the user is the owner of the activity or the ownership mode is set to Book or Mixed, then Oracle CRM On Demand changes the activity.
- If the user changes an appointment in the Microsoft Outlook calendar, and if the user is not the owner of the activity and the ownership mode is Owner and Team, then Oracle CRM On Demand Desktop does not change the organizer appointment in Microsoft Outlook. It is not necessary to synchronize the appointment. Oracle CRM On Demand Desktop does not update the Oracle CRM On Demand Desktop activity.
- If the user deletes an appointment from the Microsoft Outlook calendar, and if the user is not the owner of the activity or the ownership mode is set to Book, then Oracle CRM On Demand Desktop removes the user from the participant list. If the activity is not synchronized with the Oracle CRM On Demand server, then Oracle CRM On Demand Desktop deletes the activity in Microsoft Outlook for each participant.

- If the user deletes an appointment from the Microsoft Outlook calendar, and if the user is the owner of the activity, then Oracle CRM On Demand Desktop removes the activity.
- If the user synchronizes an All Day event from Microsoft Outlook with the Oracle CRM On Demand Calendar, then the synchronization does the following:
    - Saves the event with a start time and end time of 12:00 A.M.
    - Uses the start and end date values that the user specifies in Microsoft Outlook.

      For a single All Day event, this behavior results in an Oracle CRM On Demand Calendar entry that includes a start time of 12:00 A.M and an end time of 12:00 A.M.
- If the user's record ownership is configured to book mode, then records with an owner value are not synchronized between Oracle CRM On Demand Desktop and Oracle CRM On Demand. Records with no book value are also not synchronized. If the record ownership is in mixed mode, then synchronization does occur. However, if no Owner ID or Book ID is provided, then the records are not synchronized unless the user is on the team. User mode visibility is the same visibility as in Version 2.0 and is the default visibility mode for this release.

For more information, see *How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Activities and Microsoft Outlook Calendar*.

How Oracle CRM On Demand Desktop Handles an Oracle CRM On Demand Activity That the User Saves, Changes, or Deletes

Microsoft Outlook internally associates an Oracle CRM On Demand activity with an appointment or task. Oracle CRM On Demand Desktop applies the following logic if a user saves, changes, or deletes an Oracle CRM On Demand activity:

- If the activity does not include an associated item, then Oracle CRM On Demand Desktop attempts to find the related Microsoft Outlook item, and then associates it with the activity.
- If the activity exists in Microsoft Outlook, then Oracle CRM On Demand Desktop links it to the corresponding Microsoft Outlook item.
- If the activity originates as Oracle CRM On Demand data, and if Oracle CRM On Demand Desktop cannot find a correlation, then it creates a new Microsoft Outlook item and associates it with the activity. The type of Microsoft Outlook item that Oracle CRM On Demand Desktop creates depends on the following value in the Activity field of the activity:
    - If the value in the Activity field is Appointments, then Oracle CRM On Demand Desktop creates an appointment.
    - If the value in the Activity field is Tasks, then Oracle CRM On Demand Desktop creates a task.

  The mapping that Oracle CRM On Demand Desktop creates between the Microsoft Outlook calendar item and the first Oracle CRM On Demand Desktop activity is the same as that described in *How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Activities and Microsoft Outlook Calendar*. However, Oracle CRM On Demand Desktop performs the following additions:
- Sets the value in the Show Time As field of the native Microsoft Outlook appointment to Busy
- Sets the appointment label to None

How Oracle CRM On Demand Desktop Handles an Appointment

This topic describes how Oracle CRM On Demand Desktop handles an appointment.

Oracle CRM On Demand Desktop can handle multiple users who attempt to synchronize the same appointment in Microsoft Outlook with the Oracle CRM On Demand database. In this situation, Oracle CRM On Demand Desktop creates a single activity in the Oracle CRM On Demand database and executes a preprocessing step when the user creates a new activity in the client:

- If a duplicate record exists in the Oracle CRM On Demand database, then Oracle CRM On Demand Desktop treats the insert operation as a conflict and silently merges it with the activity created by user in the client.
- If a duplicate record does not exist in the Oracle CRM On Demand database, then Oracle CRM On Demand Desktop proceeds with the insert operation in the Oracle CRM On Demand database.

For more information, see *How Oracle CRM On Demand Desktop Uses Natural Keys to Identify a Duplicate Activity*.

If the insert operation is not successful, then no rows on the mapping table are created. A native ID is the unique identifier of the record in Microsoft Outlook.

A ConversationId is an attribute of an appointment record that is used by Oracle CRM On Demand Desktop to link Microsoft Outlook items, for example, an appointment to a shared activity that is synchronized from Oracle CRM On Demand. Activity and related PIM items both have a ConversationId property with the same value. Oracle CRM On Demand Desktop populates this ID when a user creates a new appointment and adds the appointment to a related activity when the PIM item is shared. When meeting invites are sent for this appointment, the ConversationId value is attached to the invites. In this way, every meeting attendee has the same ConversationId.

How Oracle CRM On Demand Desktop Correlates an Oracle CRM On Demand Activity with PIM Data in Microsoft Outlook

When Oracle CRM On Demand Desktop synchronizes an Oracle CRM On Demand activity with Microsoft Outlook, it attempts to find the personal information manager (PIM) data that resides in Microsoft Outlook, which corresponds to the activity. PIM data is a calendar event, appointment, task or email. If Oracle CRM On Demand Desktop finds this item, then it shares it and correlates it with the Oracle CRM On Demand activity. Oracle CRM On Demand Desktop performs this correlation for each meeting where the Activities value of the Oracle CRM On Demand activity is Calendar and Activities. To perform the correlation, it uses the following keys:

1. Key 1:
    - The iCRMId equals the ConversationId of the calendar event.
    - The owner is the meeting organizer of the calendar event.
2. Key 2:
    - The owner is the meeting organizer of the calendar event.
    - The subject is the calendar event subject.
    - The StartTime equals the start time of the calendar event.

If the user creates an activity in Oracle CRM On Demand Desktop from a Microsoft Outlook calendar event, and if the user shares this activity with Oracle CRM On Demand, then the iCRMId field in the activity record in the Oracle CRM On Demand database contains a value.

How Oracle CRM On Demand Desktop Correlates Data If Oracle CRM On Demand Desktop Is Installed

The following sequence describes how Oracle CRM On Demand Desktop uses key 1:

1. User 1 does the following:

   a. Creates a meeting in Microsoft Outlook
   b. Shares this meeting with Oracle CRM On Demand
   c. Sends the meeting request to User 2

   In this situation, Oracle CRM On Demand Desktop creates an Oracle CRM On Demand activity. To populate the value in the iCRMId field in this activity record, it uses the value from the ConversationId field of the meeting. This value is unique for this meeting. Oracle CRM On Demand Desktop uses this same value to identify the meeting organizer and all the participants for this meeting.
2. User 1 synchronizes this activity, and Oracle CRM On Demand adds it to the Oracle CRM On Demand database.
3. User 2 sets the user preference to not create a PIM item automatically, which Oracle CRM On Demand Desktop will share with Oracle CRM On Demand. If user 2 receives the meeting invitation from user 1, then Oracle CRM On Demand Desktop does not share the meeting for this user in the calendar and does not create an Oracle CRM On Demand activity.
4. When User 2 synchronizes, Oracle CRM On Demand Desktop synchronizes the activity that it added in Step 2 with Microsoft Outlook. It uses the find_ol_item function to find the Microsoft Outlook item that corresponds to this activity. It finds the unshared meeting because the following situations are true:

   - This meeting contains the same ConversationId field that the iCRMId field of the Oracle CRM On Demand activity contains.
   - This meeting contains the same meeting organizer that the Activity Owner field of the Oracle CRM On Demand activity contains.

   If the meeting attendee synchronized the activity from the Oracle CRM On Demand server before this attendee receives an invitation, and if this attendee sets the preference in the Options dialog box to not share new PIM items, then Oracle CRM On Demand Desktop uses the find_proxy_item function to find the Oracle CRM On Demand activity. If Oracle CRM On Demand Desktop finds this activity, then it shares the meeting with Oracle CRM On Demand.

How Oracle CRM On Demand Desktop Correlates Data If Oracle CRM On Demand Desktop Is Not Installed

Assume the following situation is true:

- To track activities, a user uses Oracle CRM On Demand and Microsoft Outlook.
- This user has not installed Oracle CRM On Demand Desktop.
- This user enters activities in Oracle CRM On Demand and Microsoft Outlook.
- The user has an activity in Oracle CRM On Demand. The user also has an appointment in the Microsoft Outlook calendar that matches this activity. This activity and this appointment each include the same subject, start date, and activity owner.
- The user installs Oracle CRM On Demand Desktop and then synchronizes.

In this situation, Oracle CRM On Demand Desktop cannot use Key 1, described in Step 1 in the topic *How Oracle CRM On Demand Desktop Correlates an Oracle CRM On Demand Activity with PIM Data in Microsoft Outlook*, because the Oracle CRM On Demand activity does not include a value in the iCRMId field.

How Oracle CRM On Demand Desktop Uses Natural Keys to Identify a Duplicate Activity

Oracle CRM On Demand Desktop uses natural keys to detect a duplicate between Microsoft Outlook data and Oracle CRM On Demand data. Oracle CRM On Demand Desktop defines the following natural keys for an activity:

- Activity Owner and iCRMId, which matches the ConversationId of an appointment
- Activity Owner and Description, which matches the subject of the appointment and start date

Oracle CRM On Demand Desktop uses these keys to query the Oracle CRM On Demand database. This query determines if a duplicate exists for this activity in the Oracle CRM On Demand database. The following is an example of the natural keys that Oracle CRM On Demand Desktop might define in the connector_configuration.xml file:

```
<natural_keys>
 <natural_key>
 <field>iCRMId</field>
 </natural_key>
 <natural_key>
 <field>Subject</field>
 <field>StartTime</field><!--Appointment-->
 <field>OwnerId</field>
 </natural_key>
 <natural_key>
 <field>Subject</field>
 <field>DueDate</field><!--Task-->
 <field>OwnerId</field>
 </natural_key>
</natural_keys>
```

For more information, see *Files in the Customization Package*.

How Oracle CRM On Demand Desktop Handles a Recurring Microsoft Outlook Appointment

If you have a recurring calendar event in Microsoft Outlook when you synchronize with Oracle CRM On Demand Desktop, then this recurring event is transformed into a series of individual nonrecurring Oracle CRM On Demand appointments.

When Oracle CRM On Demand Desktop transforms the recurring calendar activity, it also includes any modifications and deletions made to these occurrences, known as exceptions.

However, there is a limit to the number of individual Oracle CRM On Demand appointments that can be created to represent a recurring Microsoft Outlook calendar activity. Even if the end date on the recurring appointment in Microsoft Outlook is specified, then Oracle CRM On Demand Desktop limits the number of instances of a recurring activity, depending on the recurrence pattern.

The maximum number of occurrences for the individual appointment activities created in Oracle CRM On Demand Desktop depend on the recurrence pattern of a Microsoft Outlook calendar:

- **Daily.** 60 occurrences
- **Weekly.** 26 occurrences
- **Monthly.** 12 occurrences
- **Yearly.** 5 occurrences

You can modify the recurring appointment in Microsoft Outlook and the individual appointments in Oracle CRM On Demand. However, you cannot modify the list of participants in a single appointment occurrence in Microsoft Outlook.

If you modify a recurring appointment that is shared with Oracle CRM On Demand in Microsoft Outlook, then Oracle CRM On Demand Desktop does the following:

- If a single occurrence of a recurring appointment is modified, then Oracle CRM On Demand Desktop applies the modification to the corresponding individual appointment activity created in Oracle CRM On Demand for this appointment.
- If the entire series of a recurring appointment is modified, then Oracle CRM On Demand Desktop applies the modification to all individual appointment activities created in Oracle CRM On Demand for this appointment.
- If a single instance of a recurring appointment is deleted in Microsoft Outlook, then Oracle CRM On Demand Desktop deletes the corresponding appointment activity in Oracle CRM On Demand after synchronization.
- If all occurrences of the recurring appointment are deleted in Microsoft Outlook, then Oracle CRM On Demand Desktop deletes all appointment activities on Oracle CRM On Demand.

When an Oracle CRM On Demand Desktop user who is the participant of a recurring meeting has the default preference to automatically share new Microsoft Outlook activities with Oracle CRM On Demand, the following happens:

- When the meeting request arrives in the meeting participant's Inbox, the number of Oracle CRM On Demand appointment activities that are created corresponds to the maximum number of supported instances for the recurrence pattern.
- If the meeting participant declines the meeting request, then the Oracle CRM On Demand appointment activities created in the participant's Microsoft Outlook are deleted.
- If the meeting participant declines the meeting after the individual appointment activities have been synchronized with Oracle CRM On Demand, then the participant is removed from the list of users on all appointment activities.
- When the meeting participant modifies a single occurrence, or the entire series of the recurring Microsoft Outlook meeting, the modification is not applied on the Oracle CRM On Demand appointment activity.
- When the meeting participant deletes a single occurrence, the modification is not applied on the corresponding Oracle CRM On Demand appointment activity.

If the user modifies an appointment activity in Oracle CRM On Demand that corresponds to a single occurrence of the recurring Microsoft Outlook meeting shared with Oracle CRM On Demand, then Oracle CRM On Demand Desktop behaves in the following way:

- If the appointment activity is modified, then the corresponding occurrence of the meeting will be updated in Microsoft Outlook after synchronization for all the meeting participants.
- If the appointment activity is deleted, then the corresponding occurrence of the meeting will be deleted in Microsoft Outlook after synchronization for all the meeting participants.

How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Task

This topic describes how Oracle CRM On Demand Desktop handles data for a native Microsoft Outlook task. For more information, see *How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Data and Microsoft Outlook Data*.

If the user changes or saves a native Microsoft Outlook task and the activity type is set to Task, then Oracle CRM On Demand Desktop does the following work:

- If the user saves a new Microsoft Outlook task that is shared, then Oracle CRM On Demand Desktop creates a new Oracle CRM On Demand activity.
- If the user changes a native Microsoft Outlook task, then Oracle CRM On Demand Desktop changes the corresponding Oracle CRM On Demand activity.
- If the user deletes a native Microsoft Outlook task, and if the user:
  - Is the owner of the activity, then Oracle CRM On Demand Desktop deletes the corresponding Oracle CRM On Demand activity.
  - Is not the owner of the activity, then Oracle CRM On Demand Desktop removes the user from the activity team. Oracle CRM On Demand Desktop does not delete the corresponding Oracle CRM On Demand activity.

How Oracle CRM On Demand Desktop Handles a Microsoft Outlook Email Message

Oracle CRM On Demand Desktop handles a Microsoft Outlook email message in the following ways:

- Saves the email message as an Oracle CRM On Demand activity and sets the activity type to Email. The customization package specifies the type of activity.
- Sets the Activity value to Task.
- Creates one Oracle CRM On Demand activity for each Microsoft Outlook email message that is created.
- Allows the user to link the Oracle CRM On Demand activity to an Oracle CRM On Demand record. For more information, see *How a User Can Link an Oracle CRM On Demand Record to a Microsoft Outlook Record*.
- Saves the first email message as an .msg attachment along with the activity that is created.

If the user deletes the source email message or moves it to a new folder, then Oracle CRM On Demand Desktop does not change the activity. Deleting or modifying the activity does not affect the source email. For more information, see *How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Data and Microsoft Outlook Data*.

How Oracle CRM On Demand Desktop Handles Oracle CRM On Demand Data with Manual Email Processing

The user starts manual email processing from the email form. For more information, see *How a User Can Link an Oracle CRM On Demand Record to a Microsoft Outlook Record*.

How Oracle CRM On Demand Desktop Handles Items If the User Removes Oracle CRM On Demand Desktop

If the user removes Oracle CRM On Demand Desktop, then it removes all Oracle CRM On Demand data. How Oracle CRM On Demand Desktop handles a shared Microsoft Outlook item when the user removes Oracle CRM On Demand Desktop depends on if the item is Microsoft Outlook data or Oracle CRM On Demand data, and on the type of object. Oracle CRM On Demand Desktop handles objects in the following ways:

- **Shared appointments.** Oracle CRM On Demand Desktop removes appointments that originate in Oracle CRM On Demand from the Microsoft Outlook calendar. For a Microsoft Outlook appointment, it removes any Oracle CRM On Demand activities that are related to the Microsoft Outlook appointment. The appointment no longer displays as shared and no contextual Oracle CRM On Demand data is related to the appointment.
- **Shared contacts.** Because Oracle CRM On Demand Desktop cannot determine whether a contact is Microsoft Outlook data or Oracle CRM On Demand data, Oracle CRM On Demand Desktop removes all shared contacts from the user’s Contact folder in Microsoft Outlook. It is recommended that you back up or unshare contacts that the user must preserve before removing Oracle CRM On Demand Desktop.
- **Shared emails.** Oracle CRM On Demand Desktop removes Oracle CRM On Demand activities that are associated with shared emails so they no longer display as shared in Microsoft Outlook and so that Microsoft Outlook does not display any contextual data.
- **Shared tasks.** Oracle CRM On Demand Desktop handles tasks in the same way that it handles appointments. Oracle CRM On Demand Desktop removes tasks that originate in Oracle CRM On Demand from Microsoft Outlook. Oracle CRM On Demand Desktop does not remove a native Microsoft Outlook task. Microsoft Outlook does not display the task as a shared task, and it does not display any Oracle CRM On Demand data that is related to the task.

An Unshared Item Is Not Affected If the User Removes Oracle CRM On Demand Desktop

If the user removes Oracle CRM On Demand Desktop, then an unshared item is not affected. If the user shares an item in Microsoft Outlook, unshares it, and then synchronizes with the Oracle CRM On Demand server before the user removes Oracle CRM On Demand Desktop, then the item is not shared. This item is not affected if the user subsequently removes Oracle CRM On Demand Desktop. This situation occurs because Oracle CRM On Demand Desktop deletes only Oracle CRM On Demand data and extensions to Microsoft Outlook that you use with Oracle CRM On Demand Desktop.

How a User Can Link an Oracle CRM On Demand Record to a Microsoft Outlook Record

Oracle CRM On Demand Desktop allows the user to change linked values. For example, to choose Oracle CRM On Demand records to link with the email, the user can use the email form, and then perform the following work:

- Use the autocomplete feature when the user types in the field.
- Use the autosuggest feature when the user clicks Contact, Account, or Opportunity on the Extension Bar of the email form.
- Choose an item from an Oracle CRM On Demand control on any shared Microsoft Outlook item:
  - The Oracle CRM On Demand control calls the appropriate dialog box that allows the user to choose one or more records.
  - The dialog box supports creating a new record so long as the permissions on the source dialog box allow that operation.

In another example, Oracle CRM On Demand Desktop allows the user to link an Oracle CRM On Demand activity to one of the following Oracle CRM On Demand records:

- One account
- One lead
- Multiple users
- A book

6 How Oracle CRM On Demand Desktop Synchronizes Data

How Oracle CRM On Demand Desktop Synchronizes Data

This chapter describes how Oracle CRM On Demand Desktop synchronizes data. It includes the following topics:

- *About Local Synchronization*
- *How Oracle CRM On Demand Desktop Synchronizes Data with Oracle CRM On Demand*
- *Factors That Determine What Data an Oracle CRM On Demand Desktop User Can Access*
- *How Oracle CRM On Demand Desktop Handles a Conflict During Synchronization*

About Local Synchronization

Local synchronization allows users to synchronize changes to records that occurred in Microsoft Outlook with corresponding records in Oracle CRM On Demand, and to synchronize changes made in CRM On Demand with those same records back to Microsoft Outlook.

In local synchronization, only the records modified in Microsoft Outlook are synchronized with Oracle CRM On Demand. To synchronize all the records in Microsoft Outlook, you must run full synchronization. Oracle CRM On Demand Desktop allows users to switch between full synchronization and local synchronization.

During each local synchronization, Oracle CRM On Demand Desktop does the following:

- Builds a snapshot of records modified in Oracle CRM On Demand Desktop and compares them with the same records in Oracle CRM On Demand. If there are no collisions, then the changes are saved in Oracle CRM On Demand. Otherwise, you are prompted to resolve the synchronization collisions.
- If you have created new records in Oracle CRM On Demand Desktop, then local synchronization adds them to Oracle CRM On Demand.
- Deletes records you have deleted in Oracle CRM On Demand Desktop in Oracle CRM On Demand. If these records have been modified in Oracle CRM On Demand, then a collision occurs.

Note: The default synchronization in Oracle CRM On Demand Desktop is local synchronization. If you double-click the CRM Desktop icon in the system tray, then local synchronization launches. However, clicking the Synchronize Now button in the Resolve Conflicts tab in the Control Panel always launches full synchronization, regardless of the default synchronization option.

For more information on Windows registry parameters that affect local synchronization, see *Windows Registry Parameters That Affect Local Synchronization*.

How Oracle CRM On Demand Desktop Synchronizes Data with Oracle CRM On Demand

This topic describes how Oracle CRM On Demand Desktop synchronizes data between Microsoft Outlook and Oracle CRM On Demand. It includes the following information:

- *How Oracle CRM On Demand Desktop Synchronizes Data During the Initial Synchronization*
- *How Oracle CRM On Demand Desktop Synchronizes Data During an Incremental Synchronization*
- *How Oracle CRM On Demand Desktop Synchronizes Oracle CRM On Demand Data*
- *How Oracle CRM On Demand Desktop Manages Synchronization Duration*
- *Situations Where Oracle CRM On Demand Desktop Redownloads Oracle CRM On Demand Data*

How Oracle CRM On Demand Desktop Synchronizes Data During the Initial Synchronization

An initial synchronization is a type of synchronization that occurs in the following situations:

- Immediately after the user installs Oracle CRM On Demand Desktop.
- If you make a metadata change to the user that includes a change to the customization package.
- If the options in the login dialog box change, for example, if the user name changes or the Oracle CRM On Demand URL changes.

The purpose of the initial synchronization is to initialize the Microsoft Outlook data storage with the Oracle CRM On Demand data that is available to the user. Oracle CRM On Demand Desktop downloads the customization package to Microsoft Outlook the first time the user synchronizes metadata with Oracle CRM On Demand.

How Oracle CRM On Demand Desktop Handles Errors During Metadata Download

If the current version of Oracle CRM On Demand Desktop is not compatible with the downloaded customization package, then Oracle CRM On Demand Desktop does not apply the package. Instead, it displays an error message to notify the user, and then records an entry in the general log file:

```
Logs\GeneralLog\log.####.txt
```

where `####` are the assigned log numbers.

For information about where Oracle CRM On Demand Desktop stores this log file, see *About Files, File Locations, and Profiles*.

If an error occurs while the customization package is downloaded, then Oracle CRM On Demand Desktop displays an error message in the system tray. This error notifies the user that the customization package changed but Oracle CRM On Demand Desktop cannot download it because of certain errors. The problem might be because the user does not possess the required privilege on the Oracle CRM On Demand server. In this situation, the user must contact the system administrator to acquire the necessary privileges and then restart the synchronization. Also, the problem might be due to the fact that the user does not have the correct package published to that user's role, or the customization package might not have all the files needed. You must also check that the ZIP file for every customization package contains all the required metadata files, as you can have different packages for different roles.

How Oracle CRM On Demand Desktop Synchronizes Data During an Incremental Synchronization

An incremental synchronization is a synchronization session that occurs any time after the initial synchronization. To determine the differences that exist in the data that is available to the user, Oracle CRM On Demand Desktop compares data in the Microsoft Outlook data storage to data in the Oracle CRM On Demand database. Oracle CRM On Demand Desktop then does the following:

- Inserts, updates, or deletes data on the Oracle CRM On Demand server according to changes that occurred in Microsoft Outlook since the prior synchronization.
- Inserts, updates, or deletes data in Microsoft Outlook according to changes that occurred on the Oracle CRM On Demand server since the prior synchronization.

Oracle CRM On Demand Desktop does this work for each difference until all data in the Microsoft Outlook data storage is synchronized with data in the Oracle CRM On Demand Desktop database. In all situations, the user works with data locally in Microsoft Outlook and Oracle CRM On Demand Desktop sends those changes to the Oracle CRM On Demand server during an incremental synchronization but not at the same time that it makes the change in Microsoft Outlook. Depending on the frequency of the process, a change might not appear on the server immediately.

How Oracle CRM On Demand Desktop Handles Changes to Login Credentials

If the user name or the server URL changes, then Oracle CRM On Demand Desktop clears the data it has downloaded from Oracle CRM On Demand. Oracle CRM On Demand Desktop does this to remove any personal user data that might exist and to provide the user with an opportunity to synchronize data. Before Oracle CRM On Demand Desktop begins the reinitialization, it displays a warning message to the user that any data that is not synchronized might be lost. If the user agrees to proceed, then the following occurs:

1. Oracle CRM On Demand Desktop removes the current customization.
2. The user logs in with new credentials.
3. Oracle CRM On Demand Desktop downloads the package from the Oracle CRM On Demand server and then starts the CRM On Demand Desktop Assistant.

How Oracle CRM On Demand Desktop Synchronizes Oracle CRM On Demand Data

Oracle CRM On Demand Desktop synchronization of data with CRM On Demand is based on the configuration of the customization package. The customization package contains configuration files, which include objects, object fields, and object relationships.

To synchronize Oracle CRM On Demand data, for example, opportunities or accounts, Oracle CRM On Demand Desktop does the following:

1. Verifies the user login and connection to Oracle CRM On Demand.
2. Checks the version to ensure Oracle CRM On Demand Desktop is compatible with the current version of Oracle CRM On Demand.

3. Checks the version of Oracle CRM On Demand Desktop to ensure it has the latest version of the customization package.
4. Determines the number of records of each type of record, such as opportunities or accounts, which are records that are visible to the user through the ownership or team membership view mode, or book-based visibility for the record type. When book-based visibility is set for a record type, the Oracle CRM On Demand Desktop synchronizes records of that type, which are associated with the user specified book. Data filters are applied to the top of the records that are visible to the user, and records are then restricted by the filter configuration. The basic logic is to find all the records that are visible on the server side, compare them with the client side, and calculate the difference between the server and the client. All synchronization uses this logic.
5. Gets the values of the change keys for all synchronization objects that are enabled, such as opportunities or accounts, for example, the record ID and the last updated date in the Oracle CRM On Demand database. The change key by default includes the record Id and the time the record was last updated in the Oracle CRM On Demand database.
6. Compares the set of IDs and timestamps in Microsoft Outlook with the set of IDs and timestamps on the Oracle CRM On Demand server to do the following:
   - Determine the differences that exist between the data sets for insert, update, and delete operations.
   - Identify conflicts and create a log entry in the synchronization conflict list for any conflict.
7. For each difference, Oracle CRM On Demand Desktop performs one of the following operations in Microsoft Outlook or on the Oracle CRM On Demand server:
   - **Insert.** Query the Oracle CRM On Demand database to get the details of the new record and then insert the appropriate item in Microsoft Outlook.
   - **Update.** Query the Oracle CRM On Demand database to get the details of the updated record and then update the appropriate item in Microsoft Outlook. Note that an Oracle CRM On Demand update overwrites all fields in the corresponding Microsoft Outlook item, not just the updated fields.
   - **Delete.** Delete the appropriate item in Microsoft Outlook.
   - **Microsoft Outlook insert.** Use the user key that is defined in the metadata to query the Oracle CRM On Demand database, and then do one of the following:
     - If it does not find a match, then it inserts the appropriate record in the Oracle CRM On Demand database, and then queries the Oracle CRM On Demand database to get the record ID and timestamp.
     - If it does find a match, then it returns a deduplication conflict, a situation when two records, one in Microsoft Outlook and the other in Oracle CRM On Demand, have the same values for the fields acting as user keys. This deduplication conflict is not an error. After the user resolves the conflict by selecting the correct values or object instance, the object is synchronized. If the user does not resolve the issue, then synchronization continues, but the object involved in the conflict is not synchronized.
   - **Microsoft Outlook update.** update. Use the user key that is defined in the metadata to query the Oracle CRM On Demand database, and then do one of the following:
     - If it does not find an update for the modification number of the record, then it updates the appropriate record in the Oracle CRM On Demand database and then queries the Oracle CRM On Demand database to get the record Id and updated timestamp.
     - If it does find an update for the modification number, then it returns an update conflict, a situation when two records, one in Microsoft Outlook and the other in Oracle CRM On Demand, have the same modification number, that is, they are changed in both places. This update conflict is not an error. After the user resolves the conflict by selecting the correct values or object instance, this object is synchronized. If the user does not resolve the issue, then synchronization continues, but the object involved in the conflict is not synchronized. In this situation, Oracle CRM On Demand Desktop makes the change in the Oracle CRM On Demand database during the update operation.
   - **Microsoft Outlook delete.** Delete the appropriate record in the Oracle CRM On Demand database.

8. If a conflict occurs, then Oracle CRM On Demand Desktop does the following:
    a. Updates the synchronization issues and the Syncdump.xml log file that describes conflicts on the client.
    b. Prompts the user to choose which changes to keep in each of the following situations:
        - Update the record in Microsoft Outlook and on the Oracle CRM On Demand server.
        - Update the record in one data set and delete the record in the other data set.
9. Repeat Step 4 for each additional Oracle CRM On Demand data object that requires synchronization.

How Oracle CRM On Demand Desktop Manages Synchronization Duration

Several factors determine the duration of a synchronization, such as the amount of data that is available to the user, network bandwidth, server performance, client performance, and so on. To shorten this duration, you or the user can do the following:

- You can modify how the synchronization in Oracle CRM On Demand Desktop is configured. For more information, see *Customizing Synchronization*.
- The user can adjust settings through the synchronization filter dialog box. For more information, see *Filters Reduce the Number of Oracle CRM On Demand Records That Are Available in Oracle CRM On Demand Desktop*.
- The user can adjust the Record Set tab settings to reduce the number of records to be synchronized.

The duration of an incremental synchronization session is typically shorter than for an initial synchronization because Oracle CRM On Demand Desktop downloads all objects during an initial synchronization. However, during an incremental synchronization, Oracle CRM On Demand Desktop downloads only those objects that are changed.

Situations Where Oracle CRM On Demand Desktop Redownloads Oracle CRM On Demand Data

Oracle CRM On Demand Desktop redownloads the Oracle CRM On Demand data in any of the following situations:

- The customization package has changed.
- The package update for the user involves a data schema change.
- The user logs in as a different user.
- There is a problem with the data structure. For example, assume the user deletes the Opportunities folder and then removes this deletion from the Deleted Items folder. If the user restarts Microsoft Outlook, then Oracle CRM On Demand Desktop does the following:
    - Informs the user that there is a problem with the data structure
    - Removes the data structure
    - Installs a new data structure

If Oracle CRM On Demand Desktop must redownload the Oracle CRM On Demand data, then it does the following:

1. Removes all Oracle CRM On Demand data, such as accounts, opportunities, shared contacts, and activities
2. Removes all shared appointments and tasks that the user created for an Oracle CRM On Demand activity
3. Removes the custom data structure that it previously sent to Microsoft Outlook data storage, for example, it removes all custom folders in the user mailbox.

4. Downloads the new data.
5. Starts a new, initial synchronization session to reenter the appropriate Oracle CRM On Demand data in the Microsoft Outlook data storage.

What Happens If the Customization Package Has Changed?

During synchronization, Oracle CRM On Demand Desktop determines if the customization package for the user who is currently logged in has changed in such a way that it must reinstall the data structure. The following changes in the data structure of the customization package can cause this situation:

- An object is added to or deleted from the mapping scheme
- A field is added to an existing object or an existing field is modified

If the customization package has changed, then Oracle CRM On Demand Desktop displays a prompt that is similar to the following:

```
A new customization package is available. Select Yes to download the new version.
During the download of the new version, recent unsynchronized changes will be
removed, and an initial sync will be performed.
```

What Happens If the Customization Package Has Changed but the Data Structure Has Not Changed?

If, during synchronization, Oracle CRM On Demand Desktop determines that the customization package for the user who is currently logged in has changed in such a way that there is no change to the data structure, then Oracle CRM On Demand Desktop downloads and installs the new package and informs the user about this download. The synchronization continues. A modification to a security rule is an example of where the package has changed, but the data structure has not changed. In this situation, Oracle CRM On Demand Desktop does not start a new, initial synchronization.

Situations Where Local Data Might Be Lost

To prevent losing data due to a reinstallation of the data structure, Oracle CRM On Demand Desktop must use the current customization package to upload local data to the Oracle CRM On Demand server. The exception to this requirement occurs if the package is changed locally. In this situation, Oracle CRM On Demand Desktop cannot use the current, flawed package to synchronize data. Data for the user can be lost in the following situations:

- If data exists that Oracle CRM On Demand Desktop cannot upload to the Oracle CRM On Demand server because the permissions for the user have changed.
- If data exists for a user that is associated with synchronized items but is stored in fields that are not synchronized. Oracle CRM On Demand Desktop completely replaces the record in Microsoft Outlook with the record that it downloads from the Oracle CRM On Demand server. It erases any local data.

How Oracle CRM On Demand Desktop Prevents Data Loss If the User Deletes Customization Package Files

If Microsoft Outlook is open, then the user cannot delete any customization package files. For example, if the user attempts to use Windows Explorer to delete files from the following directory, then Windows Explorer does not allow the deletion:

```
C:\Documents and Settings\username\Application Data\Oracle\CRM OnDemand
Desktop\Profile\data
```

If Microsoft Outlook is not running, then the user can use Windows Explorer to delete customization package files. However, if the user subsequently starts Microsoft Outlook, then Oracle CRM On Demand Desktop restores the customization package files from local storage. This local storage is a PST folder or an Exchange mailbox.

How Oracle CRM On Demand Desktop Handles Data During a Connection Failure

An internet or network-connection failure that occurs during synchronization can interrupt the synchronization. An interruption does not cause data loss or corruption. Synchronization can proceed from the last step that Oracle CRM On Demand Desktop executed successfully before the interruption.

How Oracle CRM On Demand Desktop Handles Server Busy Responses

If the client starts a synchronization process (manually or scheduled) and the Oracle CRM On Demand server returns a busy response, then the client waits an amount of time (default of 10 minutes, that is configurable in the Windows registry) before it automatically attempts to synchronize data again. The client continues to attempt synchronization until it is successful, or it reaches the retry limit. The default is 144 retries, which you can configure in the Windows registry. In addition, consider the following:

- This attempt to resynchronize applies to incremental and initial synchronizations only.
- There is no automatic retry on a local synchronization because local synchronization occurs more often and is of shorter duration. Consequently, the chances of a busy server are reduced. Additionally, if the synchronization does not complete for a day, the next incremental synchronization captures all changes.
- Users can perform a manual local synchronization if the scheduled synchronization fails.

  Depending on the number of users that attempt to synchronize at the same time, the synchronization might take some time to complete.

For more information on the Windows registry parameters that control the synchronization retry interval (SyncRetryInterval) and the synchronization retry maximum attempts (SyncRetryMaxAttempt), see *Windows Registry Parameters That Affect Local Synchronization*.

Related Topics

Overview of Oracle CRM On Demand Desktop Architectural Components

Customizing Synchronization

Factors That Determine Which Data Oracle CRM On Demand Desktop Synchronizes

This topic describes the factors that determine which data Oracle CRM On Demand Desktop synchronizes. It includes the following information:

- *How Oracle CRM On Demand Desktop Synchronizes Data Depending on the Records Visibility*
- *Factors That Determine What Data an Oracle CRM On Demand Desktop User Can Access*
- *How Differences in Data Between Microsoft Outlook and the Oracle CRM On Demand Server Affect Synchronization*
- *Modifying Synchronization Frequency*
- *How Oracle CRM On Demand Desktop Avoids Duplicate Data*

How Oracle CRM On Demand Desktop Synchronizes Data Depending on the Records Visibility

Records availability in Oracle CRM On Demand Desktop after synchronization depends on the Ownership mode in Oracle CRM On Demand, the data visibility mode in Oracle CRM On Demand Desktop, and whether the record has been created in Microsoft Outlook or Oracle CRM On Demand. For more information on each case of records availability, see the topics that discuss synchronization based on records visibility in Oracle CRM On Demand Online Help.

If the visibility of a record type is set to All records where I am owner or on the team for a record type, then the synchronization downloads all records of that type that are accessible to the user through ownership and team membership.

If the visibility of a record type is set to All records in a book, then the synchronization downloads all records of that type that are associated with the specified book. If subbooks are selected for the visibility book, then records that are visible through associations with subbooks are synchronized as well.

Factors That Determine What Data an Oracle CRM On Demand Desktop User Can Access

An Oracle CRM On Demand user can typically access only a subset of data that is available in the Oracle CRM On Demand database. This topic describes the factors that determine which data an Oracle CRM On Demand Desktop user can access. How you configure Oracle CRM On Demand Desktop determines many aspects of which data Oracle CRM On Demand Desktop synchronizes, for example:

- Synchronization objects that are configured
- Master filters that are applied
- View modes that are configured on each object.

  For more information, see *How Oracle CRM On Demand Desktop Works* and *Customizing Oracle CRM On Demand Desktop*
- Book-based visibility that is configured for a record type
- Security and other configuration that exists on the Oracle CRM On Demand server

You define this configuration before you release Oracle CRM On Demand Desktop to your users environment. In the CRM On Demand Desktop Assistant, the user can choose preset filters for a predefined filter and define personal filters. The master filters and server application metadata configuration always restrict access to some data, and the user filters apply a second layer of filtering.

Oracle CRM On Demand Desktop applies these filters during the initial synchronization and incremental synchronization. Filters are used to limit the server data that is synchronized to Microsoft Outlook. However, filters are applied to server data only, and all data that is created in Microsoft Outlook is synchronized with the server.

Users cannot modify certain filters, for example, master filters. The company administrator creates and implements certain filters, for example, a filter to prevent Oracle CRM On Demand Desktop synchronizing inactive contacts with Microsoft Outlook. Master filters can be applied to both parent and child objects.

Users can modify user synchronization filters. These filters are displayed in the Filter Records screen in the Control Panel. They can also configure prebuilt filters, also known as ***filter presets***. Several filter presets can be configured in a customization package, each with its own set of criteria for different object types. The customization package contains a predefined, default filter preset. This filter is applied by default if the user does not make any modifications when running CRM On Demand

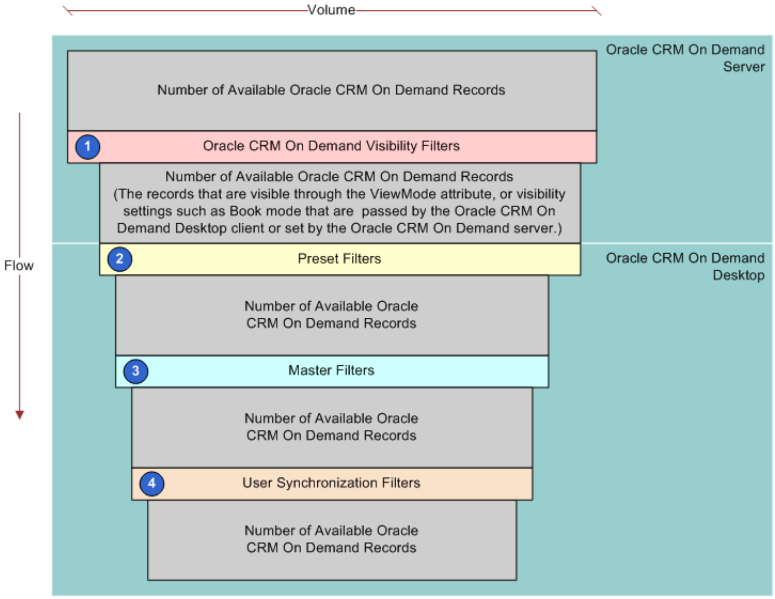
Desktop Assistant. When the user selects and modifies any preset filter, the preset filter is saved and applied until the user selects another preset filter in the Presets list.

Oracle CRM On Demand Desktop permits data access through the Book visibility in addition to the user ownership and team membership modes. It is possible to synchronize records that are visible with the user-specified book. These visibility configurations must be set for each record type on the Records Set tab of the Synchronization Control Panel of Oracle CRM On Demand Desktop.

In addition, Manager visibility allows users to see their own data and data that is directly owned by their subordinates. See Oracle CRM On Demand Online Help for more information

Filters Reduce the Number of Oracle CRM On Demand Records That Are Available in Oracle CRM On Demand Desktop

The following figure illustrates how the number of Oracle CRM On Demand records that are available to the Oracle CRM On Demand Desktop Client reduces because these records encounter each set of filters.


Volume
Oracle CRM On Demand Server
Number of Available Oracle CRM On Demand Records
1 Oracle CRM On Demand Visibility Filters
Number of Available Oracle CRM On Demand Records (The records that are visible through the ViewMode attribute, or visibility settings such as Book mode that are passed by the Oracle CRM On Demand Desktop client or set by the Oracle CRM On Demand server.)
Flow
2 Preset Filters
Oracle CRM On Demand Desktop
Number of Available Oracle CRM On Demand Records
3 Master Filters
Number of Available Oracle CRM On Demand Records
4 User Synchronization Filters
Number of Available Oracle CRM On Demand Records

The following filters reduce the number of Oracle CRM On Demand records that are available in Oracle CRM On Demand Desktop:

1. **Oracle CRM On Demand visibility filters.** Visibility rules that are configured in the Oracle CRM On Demand repository and that the Oracle CRM On Demand server applies affects data access. Because Oracle CRM On Demand Desktop integrates with the Oracle CRM On Demand server through the Web service interface, security, search specifications, and other logic that is configured at the integration or business object layer limits the data that Oracle CRM On Demand Desktop synchronizes with the client.

   The user interface configuration does not affect the results of queries or other operations that Oracle CRM On Demand Desktop performs.
2. **Preset filters.** Prebuilt filters are also known as filter presets. Several filter presets can be configured in a customization package, each with its own set of criteria for different object types. The customization package contains a predefined, default filter preset. This filter is applied by default if the user does not make any modifications when running CRM On Demand Desktop Assistant. When the user selects and modifies any preset filter, the preset filter is saved and applied until the user selects another preset filter in the Presets list.
3. **Master filters.** Master synchronization filters that are configured in the client application metadata determine which Oracle CRM On Demand data Oracle CRM On Demand Desktop synchronizes with the client. Although search

specifications on the Oracle CRM On Demand server and security settings in the Oracle CRM On Demand repository establish the first level of filtering, a set of filters that reside on the client can also restrict data that Oracle CRM On Demand Desktop downloads to the client.

4. **User synchronization filters.** The user configures synchronization filters when the user runs CRM On Demand Desktop Assistant. These filters affect which objects Oracle CRM On Demand Desktop enables for synchronization. They are determined by preset filters the user chooses or by filters that the user defines in the Filter Records Tab of the Synchronization Control Panel. To determine which data to synchronize, Oracle CRM On Demand Desktop uses this information in addition to the application configuration.

   Depending on the relationships in the data, Oracle CRM On Demand Desktop might synchronize an object that is disabled for synchronization through the Filter Records Tab. For example, if the opportunity object is enabled, but the account object is not enabled, then Oracle CRM On Demand Desktop still downloads any account data that the opportunity references. This download is required to make sure the data is complete. Also, Oracle CRM On Demand Desktop might still upload changes that the user makes in the client to the Oracle CRM On Demand server even if an object or synchronization filter is disabled. For example, if the user disables the account object and then creates an account in Microsoft Outlook, then Oracle CRM On Demand Desktop uploads the account to the Oracle CRM On Demand server.

Related Topics

Customizing How CRM On Demand Desktop Assistant Performs the Initial Synchronization

Controlling the Object Types That Oracle CRM On Demand Desktop Displays inthe Filter Records Tab

Controlling the Fields That Are Available in a Filter

Objects That Are Enabled for Synchronization

Depending on the configuration that Oracle CRM On Demand Desktop downloads for the user, a set of objects that are enabled for synchronization determine the data that Oracle CRM On Demand Desktop can synchronize. These objects are defined in the application metadata that you make available through the customization package that is available to the user. If an object is not defined in the application metadata, then Oracle CRM On Demand Desktop does not synchronize it. Application metadata also defines the field mappings that Oracle CRM On Demand Desktop uses in the synchronization. These mappings specify how Oracle CRM On Demand Desktop synchronizes objects in Microsoft Outlook and the Oracle CRM On Demand server. For more information, see *Customizing How Oracle CRM On Demand Desktop Maps Fields*.

How Differences in Data Between Microsoft Outlook and the Oracle CRM On Demand Server Affect Synchronization

For the initial synchronization, Oracle CRM On Demand Desktop downloads to Microsoft Outlook all data that resides on the Oracle CRM On Demand server that is available to the user. For an incremental synchronization, the changes that occur to data in Microsoft Outlook and on the Oracle CRM On Demand server play a large role in determining the data that Oracle CRM On Demand Desktop synchronizes. The following changes can occur:

- Data is created, updated, or deleted in Microsoft Outlook.
- Data is created, updated, or deleted on the Oracle CRM On Demand server.

In general, to determine which data is available, Oracle CRM On Demand Desktop does the following work during an incremental synchronization:

1. Uses the set of application configuration filters, user filters that are defined, and the record set configuration.
2. Identifies the differences between the data in Microsoft Outlook and the data on the Oracle CRM On Demand server. The difference is determined by a comparison of the change key values for all records that are available to the user

in Microsoft Outlook and the server. The value for the record Id resides in the ROW_ID column of the data table. The value for the time resides in the DB_LAST_UPD column of the data table. Depending on the differences, Oracle CRM On Demand Desktop changes the values in a data set to make sure the data between Microsoft Outlook and the Oracle CRM On Demand server is synchronized, for example:

  - If Oracle CRM On Demand Desktop detects a new record on the Oracle CRM On Demand server during synchronization, then it creates a corresponding record in Microsoft Outlook.
  - If Oracle CRM On Demand Desktop detects a new record in Microsoft Outlook during synchronization, then it creates a corresponding record on the Oracle CRM On Demand server.

For more information, see *How Oracle CRM On Demand Desktop Synchronizes Data During an Incremental Synchronization*.

How the Remove Local Records Synchronization Preference Affects Synchronization

To enable the Remove Local Records feature, the user can make sure the Remove Local Records Not Matching Filtering Criteria check box contains a check mark. This check box is displayed on the Filter tab of the Synchronization Control Panel. This synchronization preference allows the user to remove data that does not match a synchronization filter. If the user changes synchronization filters when this preference is enabled, even if data is not removed from the Oracle CRM On Demand server, then during synchronization Oracle CRM On Demand Desktop removes Oracle CRM On Demand data from Microsoft Outlook that falls outside of the filters. If this preference is not enabled, and if the user changes a synchronization filter, then data that was synchronized as a result of using a previous filter remains in Microsoft Outlook.

About Differences in Data Access Rules

Differences in data access rules that occur from one synchronization to the next can occur for the following reasons:

- The user downloaded a different customization package with a different configuration of synchronization objects, view modes, or master synchronization filters.
- Oracle CRM On Demand configuration has changed, which can include Role Configuration changes or Access Profiles changes.
- There can also be changes in record visibility through the owner or team or book association changes.

Modifying Synchronization Frequency

Application settings that are related to synchronization frequency determine how often and what kinds of data Oracle CRM On Demand Desktop considers for synchronization. The user can specify the frequency, or you can configure it:

- To specify the interval that Oracle CRM On Demand Desktop uses to automatically start a synchronization, the user can use the Synchronization tab of the Options dialog box in Oracle CRM On Demand Desktop. The user can also manually start a synchronization to override this configuration. The user can double-click the Oracle CRM On Demand Desktop icon in the system tray or choose the Synchronize Now option from the options menu.
- You can configure the metadata files in Oracle CRM On Demand Desktop to affect the frequency of the synchronization. This configuration determines how often Oracle CRM On Demand Desktop synchronizes each object. You can configure certain data that changes less often on the Oracle CRM On Demand server to synchronize less frequently. Example data includes List of Value and other reference data, such as users or positions. For more information, see *XML Code to Customize Synchronization*.

How Oracle CRM On Demand Desktop Avoids Duplicate Data

Oracle CRM On Demand Desktop includes metadata for the client and configuration in the Oracle CRM On Demand repository that prevents it from creating duplicate data. This configuration is in addition to the following items:

- The standard user keys included in the Oracle CRM On Demand database.
- The option to implement data deduplication that you can use to prevent duplicate data.

To define the data structures that are available to synchronize with Microsoft Outlook, Oracle CRM On Demand Desktop uses Oracle CRM On Demand object metadata definition. The object metadata definition supports the definition of a user key, which is the first additional layer of duplicate prevention.

Oracle CRM On Demand Desktop supports configuration of user keys in the metadata for the client. If Oracle CRM On Demand Desktop detects a Microsoft Outlook insert operation during synchronization, then it does the following work:

- Queries the synchronization object in the Oracle CRM On Demand database with the user key to determine whether there are any existing records that match the record being inserted.
- If Oracle CRM On Demand Desktop does not find a match, then it proceeds with the insert operation.
- If Oracle CRM On Demand Desktop finds a match, then it raises a synchronization issue that prevents the insert operation.

How Oracle CRM On Demand Desktop Handles a Conflict During Synchronization

This topic describes how Oracle CRM On Demand Desktop handles conflicts and potential duplication of Oracle CRM On Demand data that might occur during synchronization. It includes the following information:

- *How Oracle CRM On Demand Desktop Prevents Duplicate Records*
- *How Oracle CRM On Demand Desktop Handles a Synchronization Error*
- *How Oracle CRM On Demand Desktop Handles a Synchronization Conflict*

How Oracle CRM On Demand Desktop Prevents Duplicate Records

Deduplication is a feature that Oracle CRM On Demand Desktop uses to make sure that it synchronizes only a single instance of a record between Microsoft Outlook and the Oracle CRM On Demand server. Assume Oracle CRM On Demand Desktop downloads a contact to Microsoft Outlook, and then the user attempts to create another contact that contains the same name. In this situation, Oracle CRM On Demand Desktop informs the user that a contact with the same name already exists.

Assume a contact that resides on the Oracle CRM On Demand server does not also exist in Microsoft Outlook due to synchronization filters, and the user creates a contact in Microsoft Outlook that contains the same name as the contact on the Oracle CRM On Demand server. In this situation, Oracle CRM On Demand Desktop does not detect a duplicate in Microsoft Outlook and the user saves a local copy of the contact. During the next synchronization, Oracle CRM On Demand Desktop does the following work:

1. Detects the duplication error.
2. If all contact fields in both data files are the same, then Oracle CRM On Demand Desktop resolves the duplicate without interaction with the user.

3. If some of the fields are different, then Oracle CRM On Demand Desktop presents the duplicate records to the user, and then prompts the user to choose which values to keep.

The connector_configuration.xml file includes the set of objects that are configured for synchronization and the definition of the natural keys that Oracle CRM On Demand Desktop uses for deduplication. For more information, see *How the Origin of an Activity Affects Handling*.

How Oracle CRM On Demand Desktop Uses a Natural Key to Identify Duplicate Records

This topic describes how Oracle CRM On Demand Desktop uses a natural key to identify duplicates.

What Happens When Oracle CRM On Demand Desktop Inserts a Record from Microsoft Outlook to the Oracle CRM On Demand Server?

If Oracle CRM On Demand Desktop detects a potential insert operation for a record from the data store in Microsoft Outlook to the Oracle CRM On Demand database on the Oracle CRM On Demand server, then it does the following work:

1. Queries the Oracle CRM On Demand database, using natural keys in the query to identify any record in the Oracle CRM On Demand database that is a duplicate of the potential insert operation in Microsoft Outlook.
2. If Oracle CRM On Demand Desktop finds a duplicate, then it treats the transaction as a duplicate conflict. It returns the record ID of the Oracle CRM On Demand record to Microsoft Outlook so that the association between the record in the Oracle CRM On Demand database and the record in Microsoft Outlook is defined for use in incremental synchronizations. To determine whether the record exists in the Oracle CRM On Demand database before it requests an insert, Oracle CRM On Demand Desktop performs this work from Microsoft Outlook as a query to the Oracle CRM On Demand server.

What Happens When Oracle CRM On Demand Desktop Inserts a Record from the Oracle CRM On Demand Server to Microsoft Outlook

If Oracle CRM On Demand Desktop detects a potential insert operation for a record from the Oracle CRM On Demand database to the data store in Microsoft Outlook, then Oracle CRM On Demand Desktop queries the data store in Microsoft Outlook. Oracle CRM On Demand Desktop uses natural keys in the query to identify any record in the data store that is a duplicate of the potential insert operation from the Oracle CRM On Demand database.

If Oracle CRM On Demand Desktop finds a duplicate, then it treats the transaction as a duplicate conflict. For more information, see Step 2 in the topic *How Oracle CRM On Demand Desktop Uses a Natural Key to Identify Duplicate Records*.

How Oracle CRM On Demand Desktop Handles a Synchronization Error

If a top-level error occurs, then the synchronization engine stops any further processing and displays a message to the user that describes the error. Other errors might occur because of a lost connection to the server, or Oracle CRM On Demand Desktop might not be able to synchronize data due to a language or access problem. In these situations, the synchronization might be able to resume, instead of stopping. You must check the user’s connection, as well as the user's access and visibility settings. This check ensures that the user has the appropriate rights to perform all the operations required by the role. The engine also logs the message in a log file.

Oracle CRM On Demand Desktop generates the following log files, which describe synchronization errors:

- **Generic Log.** This log is enabled by default.

- **SOAP dump.**
- **Sync dump.**

You can enable the SOAP dump and Sync dump log files from the Options dialog Advanced tab, by clicking Configure Logging and Reporting.

The following table describes where the synchronization engine stores the error message log file.

Operating System	Directory
Windows 7	`C:\Users\user\AppData\Roaming\Oracle\CRM OnDemand Desktop\Profile\Logs\GeneralLog`

The following types of errors can occur in this situation:

- System error
- Resource allocation error
- General storage problem
- Application state malfunction
- Logic failure
- Connection problem
- Missing XML or JavaScript files in the customization package

If an operation failure occurs in the synchronization engine, then Oracle CRM On Demand Desktop creates a synchronization issue and then attempts to perform this operation again during the next synchronization session. The following types of errors can occur in this situation:

- Unexpected failure during an add, update, or delete operation.
- If the data of an object changed since Oracle CRM On Demand Desktop queried the object during the current synchronization session, then Oracle CRM On Demand Desktop cannot perform the update and delete operations until the next synchronization cycle. In this situation, it creates an issue that it handles in the subsequent synchronization. This synchronization creates a collision because the object was changed since the last synchronization. A collision is a situation that occurs when the same record is modified on the Oracle CRM On Demand server and in Microsoft Outlook between synchronization sessions.

How Oracle CRM On Demand Desktop Handles a Synchronization Conflict

During synchronization, Oracle CRM On Demand Desktop checks for any synchronization conflicts. A conflict occurs when:

- An item has been changed in both Oracle CRM On Demand Desktop and Oracle CRM On Demand since the last synchronization session.
- An item has been deleted in Oracle CRM On Demand Desktop and changed in Oracle CRM On Demand. If a conflict has occurred when an item is deleted in one application and changed in the other, then Oracle CRM On Demand Desktop does not select individual fields during synchronization. Either all fields are restored, or the record is deleted.

When a conflict occurs, a message and a Collisions Detected icon are displayed above the Oracle CRM On Demand Desktop icon in your system tray. Oracle CRM On Demand Desktop prompts you to open the Control Panel to resolve any conflicts.

Resolving Synchronization Conflicts

The following procedure describes how to resolve synchronization conflicts in Oracle CRM On Demand Desktop.

To resolve a synchronization conflict

1. Right-click the Oracle CRM On Demand Desktop icon, located on your system tray, and click Show Control Panel on the shortcut menu.
2. Select the Resolve Conflicts screen from the control panel.

   Resolve Conflicts is displayed in bold if there are duplicates found during synchronization. The number of duplicate conflicts is also displayed.

   The screen displays a separate record for each conflict. Each conflicting item displays the following information when you click the record to expand it:

   - **Conflict Status Color Indicator.** The conflict status color indicator is red if the conflict is unresolved and is grey if the conflict is resolved.
   - **Item Icon.** The item name is displayed for an Opportunity, Contact, Account, Activity. It also displays the name of the person who has modified the item in Oracle CRM On Demand Desktop. For example, if the contact, Margaret Burdess, was modified by William Apple, then the record is called: Margaret Burdess by William Apple
3. Click any record to view the synchronization conflict details.

   You can view only one record at a time.
4. Resolve the conflict by performing one of the following actions:

   - **Common resolution.** Click either Oracle CRM On Demand Desktop Overwrites or Oracle CRM On Demand Overwrites. All records in the list will be resolved by overwriting the record from one application with the other. For example, if you click Oracle CRM On Demand overwrites, then the Oracle CRM On Demand Desktop record will overwrite the Oracle CRM On Demand record for all conflicts in the list.
   - **Record-level resolution.** Select either the Oracle CRM On Demand Desktop Values or Oracle CRM On Demand Values option to overwrite all conflicting fields for the selected record with either Oracle CRM On Demand Desktop values or Oracle CRM On Demand values.
   - **Field-level resolution.** Select the value next to the field in either the Oracle CRM On Demand Desktop or Oracle CRM On Demand column to overwrite the selected field by the value of the appropriate application. This operation merges the records from both applications.

   Oracle CRM On Demand Desktop then displays the conflict resolution option you selected. The options are:

   - The user has not yet resolved the conflict.
   - The user has selected that the Oracle CRM On Demand Desktop values will overwrite the conflicting fields.
   - The user has selected that the Oracle CRM On Demand values will overwrite the conflicting fields.
   - The user has selected to merge Oracle CRM On Demand Desktop and Oracle CRM On Demand values to resolve the conflict.
5. After the conflict has been resolved, do one of the following:

   - Click Apply.

This step saves the resolution options for the conflict but the record will not be updated in Oracle CRM On Demand until you synchronize.

- Click Reset.

  This step restores the previously applied options to the values saved after the last time you clicked Apply.

- Click Synchronize.

  This step updates the records in Oracle CRM On Demand. Clicking Apply marks the record as resolved in your Oracle CRM On Demand Desktop, but the record will not be updated in Oracle CRM On Demand until you synchronize.

- Postpone a Conflict Resolution.

  This step postpones resolving a conflict.

- Click Close.

  This step closes the Control Panel dialog box. This does not stop the synchronization of any other records.

7 Installing Oracle CRM On Demand Desktop

Installing Oracle CRM On Demand Desktop

This chapter describes how to install Oracle CRM On Demand Desktop. It includes the following topics:

- *Roadmap for Installing Oracle CRM On Demand Desktop*
- *Process of Preparing Oracle CRM On Demand*
- *Overview of Installing Oracle CRM On Demand Desktop*
- *Process of Installing Oracle CRM On Demand Desktop*
- *Third Party Options for Installing and Configuring Oracle CRM On Demand Desktop*
- *Upgrading Oracle CRM On Demand Desktop from Version 5.0 to Version 5.1*
- *Options for Uninstalling Oracle CRM On Demand Desktop*

Roadmap for Installing Oracle CRM On Demand Desktop

To install Oracle CRM On Demand Desktop, do the following:

1. *Process of Preparing Oracle CRM On Demand*
2. *Process of Customizing Objects in Oracle CRM On Demand Desktop*
3. *Process of Installing Oracle CRM On Demand Desktop*.

Process of Preparing Oracle CRM On Demand

This process is a step in *Roadmap for Installing Oracle CRM On Demand Desktop*.

To prepare Oracle CRM On Demand for use with Oracle CRM On Demand Desktop, do the following:

1. *Preparing the Implementation Environment for Oracle CRM On Demand Desktop*
2. *Obtaining the Oracle CRM On Demand Desktop Installer*
3. *Creating and Publishing the Customization Package*
4. *Setting Oracle CRM On Demand Desktop Installer Version Locations*

Preparing the Implementation Environment for Oracle CRM On Demand Desktop

This task is a step in *Process of Preparing Oracle CRM On Demand*.

To prepare the implementation environment for Oracle CRM On Demand Desktop

1. Make sure the environment in which you intend to implement Oracle CRM On Demand Desktop supports Oracle CRM On Demand Desktop.

Before you install Oracle CRM On Demand Desktop, you must verify that your system meets the system requirements for Oracle CRM On Demand at:

http://crmondemand.oracle.com/en/products/system-requirements/index.html

You can find additional system requirements on the Oracle CRM On Demand Desktop page in Oracle CRM On Demand.

2. Make sure that the email account being used is in cached mode. Oracle CRM On Demand Desktop does not support the non-cached mode.
3. Ensure you have a valid profile ready for use before installation.

   For more information on profiles, see your Microsoft Outlook documentation.

   > **Note:** Oracle recommends having only one e-mail account for a profile in Microsoft Outlook before installing Oracle CRM On Demand Desktop.

4. Make sure that your role has the Enable CRM Desktop privilege.

   To synchronize data with the Oracle CRM On Demand server after you install Oracle CRM On Demand Desktop, you must have this privilege.

   For more information on privileges and roles, see *Oracle CRM On Demand Online Help* .
5. Ensure that the Web Services R16 Compatibility Mode setting is not selected in the Oracle CRM On Demand Company Profile.

   Oracle CRM On Demand Desktop supports only stateless Web services.

   For more information on the Oracle CRM On Demand Company Profile, see *Oracle CRM On Demand Online Help* .

Obtaining the Oracle CRM On Demand Desktop Installer

As an administrator, you can download the installation ZIP file from the Oracle Software Delivery Cloud Web site and associate the 32-bit and 64-bit versions of either the Oracle CRM On Demand Desktop or Oracle CRM On Demand Desktop Lite installer. Each installation ZIP file contains the customization package files and an EXE file, which is the installer. For more information on the customization package files, see *Creating and Publishing the Customization Package* For more information on the EXE file, see *Overview of Installing Oracle CRM On Demand Desktop*

This task is a step in *Process of Preparing Oracle CRM On Demand*.

To obtain the Oracle CRM On Demand Desktop Installer

1. Go to the following Web site:

   https://edelivery.oracle.com/
2. Click Sign In \ Register.
3. Enter your username and password, then click Sign In.
4. In the Terms and Restrictions window, select the two check boxes to accept the Electronic Delivery Trial License Agreement and the Export Restrictions, and click Continue.
5. In the Media Pack Search window, select the On Demand product pack and Microsoft Windows (32-bit) or Microsoft Windows (64-bit) platform, and then click Go.

6. In the Results pane, select Oracle CRM On Demand Desktop media pack for the version you want, and click Continue.
7. Click Download to download the ZIP file containing the client installer (EXE) and customization package for Oracle CRM On Demand Desktop.
8. Click Save to copy the Oracle CRM On Demand Desktop.5.0.0.x.exe file to your hard drive.

   You can run the EXE file to install Oracle CRM On Demand Desktop while you are offline.

Creating and Publishing the Customization Package

This task is a step in *Process of Preparing Oracle CRM On Demand*.

A customization package is a collection of XML metadata files and JavaScript files that Oracle CRM On Demand Desktop associates with a particular role. You associate a role with a customization package to determine the information that is available to the user. The metadata files customize an Oracle CRM On Demand Desktop deployment. For example, you can perform the following kinds of customization on metadata files: you can add or remove the fields that Oracle CRM On Demand Desktop synchronizes or change the layout of custom forms. For more information on customization packages, see *Files in the Customization Package*.

Note: Oracle CRM On Demand Desktop Lite does not support customization packages.

In the Package List page you can add new customization packages, and copy, edit, and delete existing packages.

Creating a Customization Package

To administer customization packages, your role must include the Manage CRM Desktop privilege and Enable CRM Desktop privilege. For administrative purposes, one free license is provided to the company.

To create a customization package

1. In the upper-right corner of any page, click the Admin global link.
2. In the Desktop Integration Administration section, click CRM Desktop Administration.
3. Click Administer Customization Packages.
4. On the Package list page, click the New Package button.
5. On the Package Edit page, complete the following fields.

Field	Comments
Package Name	Enter a name. It is recommended that you name packages according to the user role.
User Role	Select the role associated with the package.
Package Version	Enter a version number for the package. This number is for customer use only.
Attachment	Click the attachment icon, and select the package file.
Start Date	Select a start date for the package if you require a date other than today's date.

Field	Comments
End Date	If required, then select an end date for the package. Published packages without a defined end date are considered to be active indefinitely. Therefore, you must define an end date to deactivate a package record and allow a new package to be published for a role.

6. Click Save.

Alternatively you can copy an existing package and change the details to meet your requirements.

Publishing or Unpublishing Customization Packages

To make a customization package active and available to Oracle CRM On Demand Desktop users, you must publish the package.

Package records become read-only when they are published, and the value in the Status field changes from Unpublished to Published. Only one customization package can be active for a particular role at any time. Therefore, if you try to publish a package for a role with a start date that overlaps with an existing customization package for the same role, then an error message is displayed.

If it is necessary to change a metadata file in a package, then you must unpublish the package, attach the updated package file, and publish the updated package.

> **Tip:** Because users with a specified role will not be able to download their package file while the record has the Unpublished status, it is recommended that you specify an end date for the currently active package and create a new package record when changes must be made. This action ensures that a package is always available to new users and also allows for more accurate version control and tracking.

You publish a package when the updates are finished and the package is ready to download. Publishing makes a package read-only so that no more modifications can be performed on the package.

To publish or unpublish a package

1. In the upper-right corner of any page, click the Admin global link.
2. In the Desktop Integration Administration section, click CRM Desktop Administration.
3. Click Administer Customization Packages.
4. On the Package List page, click the name of the package.
5. Click the Publish button or the Unpublish button as required.

   When you publish a package, a value is generated in the Hash Value field. When you unpublish a package, the Hash Value field is empty. For more information on how a customization package works with Oracle CRM On Demand Desktop, see *Overview of Relationships Between Metadata Objects*.

Setting Oracle CRM On Demand Desktop Installer Version Locations

The Set Installer Version Location screen lists all of the Oracle CRM On Demand Desktop versions. In the Set Installer Version Location screen, you can configure the installer download location for one or more versions by entering the URL for the installer EXE files. The URL specified can be any location on your network, or on Oracle CRM On Demand Desktop using client side extensions. The versions that you configure are then available from the My Setup page.

To set the installer version location

1. Navigate to the CRM On Demand Desktop Administration screen.
2. Click the Download Location link.
3. In the Set Installer Version Location screen, do the following:

   a. For each version for which you want to configure a download location, enter the URL for the EXE files in the URL field next to the version.

   b. Click Save.

Overview of Installing Oracle CRM On Demand Desktop

An installation package is a package that is composed of a Windows Installer (EXE) file.

Oracle CRM On Demand Desktop is pre-packaged with SSO, which is embedded into the Oracle CRM On Demand Desktop .EXE file. For more information on Oracle CRM On Demand Desktop SSO, see *Administering Single Sign-On*

When you obtain the installer from the Oracle Software Delivery Cloud Web site, the installer is saved as a ZIP file. This ZIP file contains the following files:

- **Installer file.** This is the Windows Installer EXE file.
- **Package file.** This is a second ZIP file that contains metadata configuration files.
- Oracle CRM On Demand Desktop Customization Guide (PDF). This guide describes how to customize Oracle CRM On Demand Desktop.
- **Oracle CRM On Demand Desktop Administration Guide (PDF).** This guide describes how to install and configure Oracle CRM On Demand Desktop.

You can also make Oracle CRM On Demand Desktop available through third-party, deployment software that you choose. You can use the distribution criteria in these products to distribute software to any group of users, operating systems, domains, workgroups, and so on. Systems Management Server (SMS) from Microsoft is an example of deployment software. A collection is a list of users, computers, workgroups, or domains to which you must distribute the software.

You can use third-party, deployment software to perform an installation in the background or to perform a removal that uses the default installation parameters. With some deployment software, you can specify various installation parameters.

After you complete the installation and the user starts Microsoft Outlook, the CRM On Demand Desktop Assistant is displayed to help the user configure Oracle CRM On Demand Desktop. For more information, see *Customizing the CRM On Demand Desktop Assistant*.

Oracle CRM On Demand Desktop installation includes support for a flexible single sign-on system. Once Oracle CRM On Demand Desktop is installed, see *Administering Single Sign-On*, for additional information about configuring single sign-on functionality.

You can use several third-party add-ins with Oracle CRM On Demand Desktop and with Oracle CRM On Demand Desktop Lite. It is recommended that you review which third-party add-ins are supported and unsupported in *Add-In Compatibility in Oracle CRM On Demand Desktop*.

About Files, File Locations, and Profiles

Oracle CRM On Demand Desktop handles files and profiles according to the following conditions:

- To store Oracle CRM On Demand data, Oracle CRM On Demand Desktop uses a specific profile and a data folder as the default location for email delivery in the chosen profile. You can use a Microsoft Exchange Mailbox or a PST file as the location of the default email delivery for Microsoft Outlook. For more information, see *Changing the Default Mail Delivery Location in the Microsoft Outlook Profile Is Not Supported*.
- During installation, you choose the installation directory where the installer installs Oracle CRM On Demand Desktop. For more information, see *Using the Command Line to Set Optional Parameters*.
- If you configure multiple Microsoft Outlook profiles in Microsoft Outlook, then CRM On Demand Desktop Assistant only allows you to select one Microsoft Outlook profile the first time you launch Microsoft Outlook.
- If you install and then remove the Microsoft Outlook profile, then Oracle CRM On Demand Desktop remains in an installed state and continues to display in the Add and Remove Programs dialog box. In this situation, the user can remove Oracle CRM On Demand Desktop.
- Oracle CRM On Demand Desktop only supports a Microsoft Exchange Server account in cached mode. Oracle CRM On Demand Desktop uses the Microsoft Outlook .OST file to store Oracle CRM On Demand Desktop data. Microsoft Outlook synchronizes the data in the .OST file with the Microsoft Exchange Server according to a schedule. Non-cached mode is not supported by Oracle.
- If you remove the .PST file from the Microsoft Outlook profile where you install Oracle CRM On Demand Desktop, then you can still remove Oracle CRM On Demand Desktop. For more information, see *Changing the Default Mail Delivery Location in the Microsoft Outlook Profile Is Not Supported*.

Changing the Default Mail Delivery Location in the Microsoft Outlook Profile Is Not Supported

You must not change the default mail delivery location in the Microsoft Outlook profile where you install Oracle CRM On Demand Desktop.

CAUTION: If you change the default mail delivery location, then Oracle CRM On Demand Desktop will not function correctly. Changing this location is not supported.

Changes That Oracle CRM On Demand Desktop Makes During an Installation

This topic describes changes that Oracle CRM On Demand Desktop makes during installation to the file system, Windows Registry, and settings in Microsoft Outlook.

Where Oracle CRM On Demand Desktop Stores Data in the File System

Oracle CRM On Demand Desktop places most files that it requires in the following folder:

```
Application Data\Oracle\CRM OnDemand Desktop\Profile
```

where:

Application Data is an environment variable that the operating system automatically sets.

For example, in Windows 7, Oracle CRM On Demand Desktop places most files that it requires in the following folder:

```
C:\Users\username\AppData\Roaming\Oracle\CRM OnDemand Desktop
```

You can change this directory. For more information, see *Setting the Oracle CRM On Demand Desktop Installation Directory* .

The following table describes where Oracle CRM On Demand Desktop stores data in the file system when the Oracle CRM On Demand Desktop client runs on Windows 7.

Windows 7 Folder on Client Computer	Description
C:\Users\username\AppData\Roaming\Oracle\CRM OnDemand Desktop\bin	Oracle CRM On Demand Desktop saves the following information: • Binary libraries that include a .DLL extension. • Files that include common resources, such as text files, graphic files, and files that include an .XML extension. • Help files.
C:\Users\username\AppData\Roaming\Oracle\CRM OnDemand Desktop\Bin\Help	Help files.
C:\Users\username\AppData\Roaming\Oracle\CRM OnDemand Desktop\Profile	Oracle CRM On Demand Desktop saves the following information: • The data folder, which includes package files • Oracle CRM On Demand log files • Database files for the synchronization engine. These are .MDB files that contain auxiliary data.
C:\Users\username\AppData\Roaming\Oracle\CRM OnDemand Desktop\Profile\Data	XML files and JavaScript files of the customization package. For more information, see *Files in the Customization Package*.
C:\Users\username\AppData\Roaming\Oracle\CRM OnDemand Desktop\Profile\Log	The Oracle CRM On Demand log files. For more information, see *How Oracle CRM On Demand Desktop Handles Errors During Metadata Download*.
C:\Users\username\AppData\Local\Temp	When you download the customization package Oracle CRM On Demand Desktop places some files in a temp directory.

Changes That Oracle CRM On Demand Desktop Makes in the Windows Registry

Oracle CRM On Demand Desktop adds registry entries in the following registry key:

HKEY_CURRENT_USER\Software\Oracle\CRM OnDemand Desktop

These settings include the following information:

- General settings for Oracle CRM On Demand Desktop, such as login information.
- In the Logging subkey, the log file settings that it uses to configure logging behavior.

When you install Oracle CRM On Demand Desktop it registers COM classes in the Windows Registry. For more information on Windows Registry settings that Microsoft Windows requires to register COM classes, see the topic about Registering

COM Applications at the Microsoft Developer Network Web site. For more information, see *Modifying the Windows Registry to Change Oracle CRM On Demand Desktop Behavior*.

Changes That Oracle CRM On Demand Desktop Makes to Settings in Microsoft Outlook

Because Oracle CRM On Demand Desktop runs as a Microsoft Outlook add-in, it must register with Microsoft Outlook. For more information on Windows Registry settings that Microsoft Windows requires to register an add-in, see the topic about registry settings for COM add-ins at the Microsoft Developer Network Web site.

Oracle CRM On Demand Desktop adds the following items to the Microsoft Exchange folder file or personal folders file:

- Custom folders
- Custom objects
- Custom forms

For more information, see *How Oracle CRM On Demand Desktop Displays Data in Microsoft Outlook*.

Process of Installing Oracle CRM On Demand Desktop

This process is a step in *Roadmap for Installing Oracle CRM On Demand Desktop*. To install Oracle CRM On Demand Desktop, do the following:

1. *Installing Oracle CRM On Demand Desktop*
2. *Third Party Options for Installing and Configuring Oracle CRM On Demand Desktop*

Installing Oracle CRM On Demand Desktop

You can use one of the following methods to install Oracle CRM On Demand Desktop on your computer:

- *Installing Oracle CRM On Demand Desktop from My Setup*
- *Installing Oracle CRM On Demand Desktop from a Command Line*

Installing Oracle CRM On Demand Desktop from My Setup

This procedure describes how to download the EXE file and then run the EXE file to install Oracle CRM On Demand Desktop.

To run the Oracle CRM On Demand Desktop EXE file

1. If Microsoft Outlook is open, then close it.
2. In Oracle CRM On Demand Desktop, navigate to My Setup, and then Data and Integration Tools.

   Note: This step assumes the user has the Enable CRM On Demand Desktop privilege enabled. For more information on privileges and roles, see Oracle CRM On Demand Online Help.

3. Under Oracle CRM On Demand Desktop, click the Oracle CRM On Demand Desktop link.
4. In the Oracle CRM On Demand Desktop download page, under the Installation section, select either the Microsoft Windows (32-bit) or Microsoft Windows (64-bit) platform in the Select Installer drop-down list.
5. In the Installation section, click the Download CRM Desktop On Demand link to install Oracle CRM Desktop On Demand.
6. Locate the Oracle CRM On Demand Desktop.5.0.0.x.exe installation package.

7. Double click the Oracle CRM On Demand Desktop.5.0.0.x.exe file.

   The Oracle CRM On Demand Desktop.5.0.0.x.exe installation package validates the operating system version and the Microsoft Outlook version that is currently installed on the client computer. For more information, see *Preparing the Implementation Environment for Oracle CRM On Demand Desktop*.
8. In the Welcome dialog box, click Next.
9. In the License Agreement window, accept the license terms, and click Next.
10. In the Customer Information dialog box, enter the user name and the organization.
11. In the Destination Folder window, select the folder to where you want to save your Oracle CRM On Demand Desktop installation, and click Next.

    The default location for Windows 7 is: C:\Users\username\AppData\Oracle\Oracle On Demand CRM for Microsoft Outlook\
12. Click Install to start the installation.

    Oracle CRM On Demand Desktop installs in the background. The next time the user opens Microsoft Outlook, Oracle CRM On Demand Desktop starts the CRM On Demand Desktop Assistant in the client computer.
13. In the InstallShield Wizard dialog box, click Finish.
14. If you select the Launch Oracle CRM On Demand Desktop check box, then Oracle CRM On Demand Desktop finishes the installation and then does one of the following depending whether Microsoft Outlook is open:

    - **Outlook is open.** Prompts the user to apply the configuration to the current Outlook profile.
    - **Outlook is not open.** Starts Outlook and then prompts the user to apply the configuration to a profile that the user chooses.

      If you do not select the Launch Oracle CRM On Demand Desktop check box, then Oracle CRM On Demand Desktop finishes the installation and then does one of the following depending whether Microsoft Outlook is open:
    - **Outlook is open.** To start Oracle CRM On Demand Desktop, you can choose the Start menu in Microsoft Windows, choose All Programs, Oracle, and then click Launch Oracle CRM On Demand Desktop.
    - **Outlook is not open.** Oracle CRM On Demand Desktop opens the first time you open Outlook after the installation finishes.

Installing Oracle CRM On Demand Desktop from a Command Line

This topic describes how to install Oracle CRM On Demand Desktop from a command line.

To install Oracle CRM On Demand Desktop from a command line

1. Open a command line by typing `cmd.exe` in the Run window.
2. Use the following command to install Oracle CRM On Demand Desktop:

```
"C:\...\Oracle CRM On Demand Desktop.exe" /s /v"/L*v \"%temp%\CRMDesktop.log\" /
qn"
```

3. Start Microsoft Outlook.
4. Select whether you want the Oracle CRM On Demand Desktop configuration applied to the current Microsoft Outlook profile.

Third Party Options for Installing and Configuring Oracle CRM On Demand Desktop

This topic describes the third party options that are available for installing Oracle CRM On Demand Desktop. It includes the following information:

- *Customizing the CRM On Demand Desktop Assistant*
- *Installing Oracle CRM On Demand Desktop Using the Microsoft Group Policy Object*
- *Using a Microsoft SMS Package to Install Oracle CRM On Demand Desktop*
- *Using the Command Line to Set Optional Parameters*

Also, you can administer various files to change how Oracle CRM On Demand Desktop behaves. For more information, see *Changing the Behavior of Oracle CRM On Demand Desktop*.

Customizing the CRM On Demand Desktop Assistant

The CRM On Demand Desktop Assistant is a wizard that guides you through the initial setup of the Oracle CRM On Demand Desktop. The first time you start Microsoft Outlook after installing Oracle CRM On Demand Desktop, Oracle CRM On Demand Desktop starts the CRM On Demand Desktop Assistant. After you finish using the assistant, you can begin using Microsoft Outlook.

At each step, the CRM On Demand Desktop Assistant displays a dialog box where you can specify certain settings. This topic describes how you can customize the behavior of some of these dialog boxes. For more information, see *How Oracle CRM On Demand Desktop Synchronizes Data*.

Customizing How CRM On Demand Desktop Assistant Uses the Customization Package

The following table describes administrative setup work you can perform to customize how CRM On Demand Desktop Assistant registers and obtains the customization package. It lists work items in the order in which the user performs them while using the assistant. A row that does not include administrative work indicates that there is no administrative work that you can perform. In order for the assistant to start, the user must first install Oracle CRM On Demand Desktop. For more information, see *Installing Oracle CRM On Demand Desktop*.

Administrative Work	Description	For More Information
You can administer some settings for the Connection tab of the CRM Desktop-Options dialog box.	CRM On Demand Desktop Assistant checks the connection settings. If a connection is established, then the assistant continues. If a connection cannot be established, then the assistant displays the Oracle CRM On Demand Desktop-Options dialog box with the Connection tab active. The Use Internet Explorer Settings for Proxy-Server option is chosen by default. The Manual Proxy-Server Configuration option provides the user with a way to specify a proxy server. If your organization uses a	For more information on customizing the Connection tab, see *Controlling the Connection Tab*.

Administrative Work	Description	For More Information
	proxy server, then you must provide the user with the required information. The user must enter the following information: • The host name for the proxy server in the Server window • The port number in the window that displays immediately to the right of the Server window The proxy server requires a separate host name and a port number.	
None	After a network connection is established, CRM On Demand Desktop Assistant displays the Oracle CRM On Demand Desktop-Login dialog box. The user enters the user name and password in the dialog box. The user name must be the User ID of the user record in the Oracle CRM On Demand database. The user ID is the same as what is used for Oracle CRM On Demand Web access, for example, WTAKUDA. The password is the same as the password that the user uses for Oracle CRM On Demand.	For more information on logging into Oracle CRM On Demand, see *Oracle CRM On Demand Online Help* .
You can change the behavior of the Oracle CRM On Demand Desktop-Login dialog box.	You can hide the Save Password check box that Oracle CRM On Demand Desktop displays in the Oracle CRM On Demand Desktop-Login dialog box and prevent Oracle CRM On Demand Desktop from displaying the Oracle CRM On Demand Desktop-Login dialog box.	For more information, see *Changing the Login Dialog Box Behavior*, and *Modifying the Windows Registry to Change Oracle CRM On Demand Desktop Behavior*.
You can set optional parameters to specify the URL of the Oracle CRM On Demand server to which the synchronization engine connects.	CRM On Demand Desktop Assistant automatically enters the URL of Oracle CRM On Demand to the Oracle CRM On Demand server in the Server URL window, for example: http://server_name/Services/Integration/ You can specify the URL as an installation parameter through an option in the **`msiexec`** command line for Windows Installer.	For more information, see *Setting the URL for the Oracle CRM On Demand Server*.

Changing the Login Dialog Box Behavior

You can change the behavior of the Oracle CRM On Demand Desktop-Login dialog box.

To change behavior of the Oracle CRM On Demand Desktop-Login dialog box

1. To hide the Save Password check box that Oracle CRM On Demand Desktop displays in the Oracle CRM On Demand Desktop-Login dialog box, set the following Windows Registry setting to 1:

   ```
   Connector:HideSavePasswordOption
   ```

   If the user clicks Save Password in the Oracle CRM On Demand Desktop-Login dialog box, then Oracle CRM On Demand Desktop saves a copy of the password locally to the client computer. If you suppress the display of the Save Password check box, then the user must enter the password every time the user logs in to Oracle CRM On Demand Desktop.
2. To prevent Oracle CRM On Demand Desktop from displaying the Oracle CRM On Demand Desktop-Login dialog box, do the following:

a. Set the following Windows Registry setting to 1:

```
SuppressLoginDialog
```

b. Define the save_password registry key parameter in the registry editor.

c. Define the Connector:LoginName registry key parameter in the registry editor.

If you do not define the save_password parameter, then Oracle CRM On Demand Desktop requires the user to enter the password every time the user opens Microsoft Outlook and then synchronizes.

For more information, see the following topic.

How Oracle CRM On Demand Desktop Suppresses the Oracle CRM On Demand Desktop-Login Dialog Box

If you suppress the display of the Oracle CRM On Demand Desktop-Login dialog box, then Oracle CRM On Demand Desktop does the following:

- If the login, password, and URL connection parameters are present in the Windows Registry, and if save_password is present in the Windows Registry and set to 1, then Oracle CRM On Demand Desktop attempts to validate the user credentials on the Oracle CRM On Demand server.
- If the Oracle CRM On Demand server returns an error for this login, then Oracle CRM On Demand Desktop displays the Oracle CRM On Demand Desktop-Login dialog box and allows the user to attempt to log in or to cancel the login. The Oracle CRM On Demand server returns an error if it cannot validate the login credentials.
- If a connection parameter is not present in the Windows Registry, or if save_password is not present in the Windows Registry or is set to 0, then the Oracle CRM On Demand server returns a Credentials Verification Failed error. After this error, a Windows message appears in the bottom-right corner of screen. The Oracle CRM On Demand Desktop-Login dialog box is not displayed.

Customizing How CRM On Demand Desktop Assistant Performs the Initial Synchronization

After Oracle CRM On Demand Desktop installs the data structure, the second part of CRM On Demand Desktop Assistant is displayed, which prompts the user to set preferences and run the first synchronization session that downloads Oracle CRM On Demand records to Microsoft Outlook. The following table describes the administrative setup work that you can perform to customize how the assistant performs this initial synchronization. It lists the work items in the order in which the user performs them while the user runs the assistant.

Step	Description	Administrative Work
1	After CRM On Demand Desktop Assistant installs the folder structure, it presents the following choices in the Filter Records tab of the Synchronization Control Panel dialog box: • Leave the filters at their default settings. • Choose a filter from the predefined filter that Oracle CRM On Demand Desktop uses with Oracle CRM On Demand Desktop. • Specify filter settings. The user can also specify the synchronization frequency and other settings that Oracle CRM On Demand Desktop uses. As an option, you can specify the filter settings for every user and use them when you install Oracle CRM On Demand Desktop. This option allows you	You can specify the filter settings for every user and use them when you install Oracle CRM On Demand Desktop. This option allows you customize which filters and other settings that Oracle CRM On Demand Desktop displays. For more information, see *Controlling the Object Types That Oracle CRM On Demand Desktop Displays inthe Filter Records Tab*.

Step	Description	Administrative Work
	customize which filters and other settings that Oracle CRM On Demand Desktop displays.	
2	The CRM On Demand Desktop Assistant displays the Synchronization tab of the Oracle CRM On Demand Desktop-Options dialog box where the user can set the synchronization schedule. By default, Oracle CRM On Demand Desktop does the following: • Enters a check mark in the Schedule for the Automatic Synchronization Interval check box • Enters a check mark in the Show Progress During Automatic Synchronization check box • Sets the frequency slide bar to Once an Day	For more information, see *Controlling the Confirm Synchronization Tab*.
3	The CRM On Demand Desktop Assistant displays the Advanced tab of the Oracle CRM On Demand Desktop-Options dialog box, where the user can share with Oracle CRM On Demand Desktop any new native Microsoft Outlook appointments, contacts or tasks that the user creates in Microsoft Outlook. By default, Oracle CRM On Demand Desktop includes a check mark in the Appointments, Contacts, Tasks check box.	You cannot administer the settings of the advanced preferences tab.
4	The CRM On Demand Desktop Assistant displays the Oracle CRM On Demand Desktop dialog box. If SharedByDefault:NewItems registry key is set to 1, then new Microsoft Outlook items (Appointments, Task, Contacts) will be shared by default. If SharedByDefault:NewItems registry key is set to 0, then the items will be unshared by default.	For more information, see *How Oracle CRM On Demand Desktop Converts Contacts*.

After the user finishes specifying the configuration settings, Oracle CRM On Demand Desktop automatically starts the synchronization and adds content to the Oracle CRM On Demand folders. This content depends on the choices the user specifies in the CRM On Demand Desktop Assistant. After the synchronization finishes, the user can find the Oracle CRM On Demand data that Oracle CRM On Demand Desktop downloads in the corresponding Oracle CRM On Demand folders. The user can view Oracle CRM On Demand contacts that Oracle CRM On Demand Desktop downloads to the Microsoft Outlook Contacts folders. Contacts that existed in Microsoft Outlook before you installed Oracle CRM On Demand Desktop are not automatically shared with Oracle CRM On Demand Desktop. The user can use icons or group contacts to separate them from the Oracle CRM On Demand contacts according to the Shared and Not Shared attribute.

How Oracle CRM On Demand Desktop Converts Contacts

During the CRM On Demand Desktop Assistant, the following confirmation message is displayed:

Should the Oracle CRM On Demand Desktop installation process convert existing Microsoft Outlook contacts to unshared business contacts?

Depending on the option you select, the following happens:

- **If you select Yes.** Oracle CRM On Demand Desktop converts native Microsoft Outlook contacts to Oracle CRM On Demand contacts, and the user can share these Microsoft Outlook contacts with Oracle CRM On Demand. Oracle CRM On Demand Desktop does not automatically share contacts. The user must manually choose to share each contact.
- **If you select No.** The contacts remain as native Microsoft Outlook contacts. After the CRM On Demand Desktop Assistant finishes, the user can use the Actions menu to convert native contacts to Oracle CRM On Demand

contacts in an unshared state, and then manually share the contacts with Oracle CRM On Demand Desktop. If the user does not convert these contacts, then the user can share the existing Microsoft Outlook contacts with Oracle CRM On Demand Desktop at a later stage.

How Oracle CRM On Demand Desktop Handles Duplicate Contacts

During initial synchronization, Oracle CRM On Demand Desktop determines whether the user's private, pre-existing Microsoft Outlook contacts are the same as downloaded contacts from Oracle CRM On Demand. If a Microsoft Outlook contact has the same first name, last name, and email address as any contact downloaded from Oracle CRM On Demand, then these contacts are considered duplicates, and you can resolve this conflict. For more information on how Oracle CRM On Demand supports deduplication, see *How Oracle CRM On Demand Desktop Handles a Conflict During Synchronization*.

To locate and merge duplicate contacts

1. In the left pane of Microsoft Outlook, select the Contacts folder.
2. In the Contacts window, click the Manage Contact Duplicates button.

   The Manage Contact Duplicates window appears. The left pane of the Manage Contact Duplicates window lists the duplicate contacts. The right pane shows values of Microsoft Outlook and Oracle CRM On Demand attribute values. Attribute values marked with a green check mark are identical, while those that are marked with a red cross are different.

   Note: If a contact has several matching duplicate contacts in Oracle CRM On Demand, then this contact cannot be merged.

3. Select a contact from the left CRM On Demand Contact pane, then click Compare and Merge.

   This opens the Resolve Duplicate window.
4. To merge duplicate contacts, do one of the following:
   - Select a contact attribute value in the left Outlook pane, and click the right arrow icon to merge that value with the Oracle CRM On Demand attribute value.
   - Click the move all icon to merge all contact values in the Outlook pane with the Oracle CRM On Demand attribute.
5. Click Merge and Next to merge all values and to switch to the next set of duplicate contacts.
6. (Optional) If you want keep a CRM On Demand contact, and if you do not want to merge it with a Microsoft Outlook contact, then click Next Contact to skip to the next duplicate contacts.
7. Repeat Step 4 and Step 5 until you have merged all the required contacts.
8. Click Close.

To merge duplicates automatically

1. In the left pane of Microsoft Outlook, select the Contacts folder.
2. In the Contacts window, click the Manage Contact Duplicates button.

   The Manage Contact Duplicates window appears. The left pane of the Manage Contact Duplicates window lists the duplicate contacts. The right pane shows values of Microsoft Outlook and Oracle CRM On Demand attribute values. Attribute values marked with a green check mark are identical, while those that are marked with a red cross are different.
3. Click Auto-Merge, then click OK to confirm.

Installing Oracle CRM On Demand Desktop Using the Microsoft Group Policy Object

This topic describes how to use the Microsoft Group Policy Object to install Oracle CRM On Demand Desktop for each user. It includes an optional step that describes how to install it for anyone who uses the client computer. For detailed information on how to use the Microsoft Group Policy Object, see the Microsoft TechNet Web site.

To use the Group Policy Object to install Oracle CRM On Demand Desktop

1. Make sure the user runs Microsoft Outlook on the client computer at least one time.

   For more information, see *Starting Microsoft Outlook Before You Install Oracle CRM On Demand Desktop*.
2. Create an .MST transform file, using a third-party tool, such as Orca.

   This file contains the custom parameters that Oracle CRM On Demand Desktop requires to complete an installation, for example, the name of the Oracle CRM On Demand server, and so on.
3. Create the Group Policy Object as follows:

   a. Navigate to your Microsoft Active Directory environment.
   b. In the Group Policy Object Editor, create a package under the User Configuration section, not the Computer Configuration.
   c. Set the Deployment method of the package to Assigned.
   d. On the Deployment tab of the package Properties, select the Install this application at logon checkbox.
   e. On the Modifications tab, specify the network path that identifies the location of the .MST file and the .MSI file, for example:

      ```
      \\server\install\CRMDesktop\transform.mst
      ```
   f. Optional. To install Oracle CRM On Demand Desktop for anyone who uses the client computer, set the ALLUSERS property to 1 in the Property table. You set this property in the transform file that you created in Step 2.
   g. Optional. Make sure the workstations where you install Oracle CRM On Demand Desktop can access the network path to the .MST file and the .MSI file that you specified in Step e.

Using a Microsoft SMS Package to Install Oracle CRM On Demand Desktop

This topic provides guidelines on how to use a Microsoft SMS Package to install Oracle CRM On Demand Desktop for each user. It includes optional steps that describes how to install it for anyone who uses the client computer. For detailed information on how to use Microsoft SMS Package, see the Microsoft TechNet Web site.

To use a Microsoft SMS Package to install Oracle CRM On Demand Desktop

1. Make sure the user runs Microsoft Outlook on the client computer at least one time.

   For more information, see *Starting Microsoft Outlook Before You Install Oracle CRM On Demand Desktop*.
2. Extract the MSI file from the EXE package by using following command:

```
Oracle CRM On Demand Desktop.5.0.0.x.exe /b"DRIVE:\PATH"
```

where DRIVE:\PATH is the path to store resulting *.msi

3. Create an .MST transform file, using a third-party tool, such as Orca.

   This file contains the custom parameters that Oracle CRM On Demand Desktop requires to complete an installation, for example, the name of the Oracle CRM On Demand server, and so on.

4. Create a .BAT file that includes the following command:

   ```
   msiexec.exe /qn /i "\\server\install\CRMDesktop\Oracle CRM On Demand
   Desktop.5.0.0.x.exe" TRANSFORMS="\\server\install\CRMDesktop\transform.mst"
   ```

5. For the Microsoft SMS Package that contains the Oracle CRM On Demand Desktop application, choose the Data Source tab.

   This package contains the source files.

6. Select the following check box:

   Always Obtain Files from Source Directory

7. For the Program object of Oracle CRM On Demand Desktop within the package:

   a. Click the General tab.
   b. In the Command line window, specify the .bat file, and select the Run to Hidden check box.
   c. Click the Environment tab, and then enter information using values from the following table.

Window	Value
Program can run	Only when a user is logged on
Run mode	Run with user's rights
Drive mode	Runs with UNC name

Starting Microsoft Outlook Before You Install Oracle CRM On Demand Desktop

If a user starts Microsoft Outlook for the first time on a new workstation, and if a configured mail profile for Microsoft Outlook does not exist, and if you attempt a silent installation, then the installation fails. For the installation to run successfully, a configured mail profile for Microsoft Outlook must exist. Microsoft Outlook creates this profile the first time a user starts Microsoft Outlook.

The following sequence must occur:

1. User starts Microsoft Outlook.
2. User exits Microsoft Outlook.
3. You install Oracle CRM On Demand Desktop.
4. User starts Microsoft Outlook again. Oracle CRM On Demand Desktop completes the installation and enters Oracle CRM On Demand Desktop data in the lists and fields in Microsoft Outlook.

Using the Command Line to Set Optional Parameters

You can use the command line to set optional parameters that affect the installation. This topic includes the following information:

- *Abbreviating the Installation Procedure*
- *Setting the Oracle CRM On Demand Desktop Installation Directory*
- *Setting the URL for the Oracle CRM On Demand Server*
- *Setting the Microsoft Outlook Profile for Oracle CRM On Demand*

You can run the Oracle CRM On Demand Desktop.5.0.0.x.exe installation package from the DOS command line interface on the client computer. Oracle CRM On Demand Desktop supports all parameters that can be set in the Windows Installer **msiexec** command line. For more information on command line options for the Windows Installer, see the Microsoft TechNet Web site.

To use the command line to set optional parameters

1. Open a command line.
2. Go to the directory that contains the Oracle CRM On Demand Desktop.5.0.0.x.exe file, for example:

   ```
   c:\WINDOWS\system32.
   ```

3. Enter the Windows Installer command using the following format:

   ```
   msiexec.exe /I Oracle CRM On Demand Desktop.5.0.0.x.exe optional_parameter_1
   optional_parameter_n
   ```

   where:

   - optional_parameter is a parameter you can enter that Oracle CRM On Demand Desktop executes when you run the Oracle CRM On Demand Desktop.5.0.0.x.exe installation package, for example:

     ```
     msiexec.exe /I Oracle CRM On Demand Desktop.5.0.0.x.exe
     INSTALLDIR=c:\My_Custom_Directory OL_PROFILE="test PST"
     ```

     Apply the following conditions:

     - You must specify each optional parameter in the same command line after the name of the Oracle CRM On Demand Desktop.5.0.0.x.exe file.
     - To separate each optional parameter, you must enter a space without a slash (/).
     - You can arrange optional parameters in any order.

4. Press Enter.

   The Welcome dialog box of the Oracle CRM On Demand Desktop Setup wizard displays.

Abbreviating the Installation Procedure

If you do not want to install using the installation dialog boxes, then you can use the optional QR parameter. If you use the QR parameter, then the Oracle CRM On Demand Desktop.5.0.0.x.exe installation package does not display dialog boxes that require user action.

To use the QR parameter to abbreviate the installation procedure

- Append the QR parameter to the **msiexec** command, for example:

  ```
  msiexec.exe /I Oracle CRM On Demand Desktop.5.0.0.x.exe
  INSTALLDIR=c:\My_Custom_Directory OL_PROFILE="test PST" /QR
  ```

Setting the Oracle CRM On Demand Desktop Installation Directory

To change the default location of where the Oracle CRM On Demand Desktop.5.0.0.x.exe installation package saves files during installation, you can use the optional INSTALLDIR parameter. Oracle CRM On Demand Desktop.5.0.0.x.exe installs to the following directory for Microsoft Windows 7, by default:

```
C:\Users\username\AppData\Roaming\Oracle\CRM OnDemand Desktop\
```

To set the Oracle CRM On Demand Desktop installation directory

- Enter the following parameter on the **msiexec** command line anywhere after the mandatory Oracle CRM On Demand Desktop.5.0.0.x.exe name parameter:

  ```
  INSTALLDIR=directory_path
  ```

  For example:

  ```
  \Documents and Settings\username\Desktop\Oracle CRM On Demand Desktop.5.0.0.x.exe
  ```

  where:

- username is the name of the user, such as WTAKUDA.

Setting the URL for the Oracle CRM On Demand Server

To specify the URL of the Oracle CRM On Demand server to which the synchronization engine connects, you can set optional parameters.

To set the URL for the Oracle CRM On Demand server

1. Enter the following parameters on the **msiexec** command line anywhere after the mandatory Oracle CRM On Demand Desktop.5.0.0.x.exe name parameter:

   ```
   ONDEMAND_SERVER_PROTOCOL=protocol ONDEMAND_SERVER_HOST=host_name_or_address
   ONDEMAND_SERVER_PORT=server_port ONDEMAND_SERVER_COMPONENT=component_name
   ONDEMAND_SERVER_SUFFIX=request_suffix
   ```

   where:

   - protocol is http, which is the default value.
   - host_name_or_address is the computer name or IP address of the target server. This parameter is empty by default. To use a fully qualified domain name for the server_address variable, you must set the EnableFQDN parameter in the configuration (cfg) file.
   - server_port is 443 for production environments. The default value is 80.
   - component_name is empty, which is the default value.

  - request_suffix is the following default value: Services/Integration/

For example:

```
msiexec.exe /I Oracle CRM On Demand Desktop.5.0.0.x.exe
ONDEMAND_SERVER_PROTOCOL=http ONDEMAND_SERVER_HOST=sdcv440s133.ondemand.com
ONDEMAND_SERVER_PORT=80 ONDEMAND_SERVER_COMPONENT= ONDEMAND_SERVER_SUFFIX=
Services/Integration/
```

No parameters are required.

Because any information that you define in these parameters sets the parameter values in the Windows Registry, the end user is not required to set them. For example, the protocol variable of the ONDEMAND_SERVER_PROTOCOL parameter overrides the Connector:Protocol entry in the Windows Registry. For more information, see *Modifying the Windows Registry to Change Oracle CRM On Demand Desktop Behavior*. You can also use XML code to override the URL. For more information, see *XML Code to Customize Forms*.

Setting the Microsoft Outlook Profile for Oracle CRM On Demand

You can set the optional OL_PROFILE parameter in the **msiexec** command line. In the context of Oracle CRM On Demand, a Microsoft Outlook profile is the standard Microsoft Outlook concept that Oracle CRM On Demand uses to recall the email accounts and the settings that tell Microsoft Outlook where email is stored for the user. The OL_PROFILE parameter specifies the Microsoft Outlook profile where Oracle CRM On Demand is installed.

To set the Microsoft Outlook profile for Oracle CRM On Demand

- Enter the following parameter on the **msiexec** command line anywhere after the mandatory Oracle CRM On Demand Desktop.5.0.0.x.exe name parameter:

  ```
  OL_PROFILE=”my_profile"
  ```

  where:

  my_profile is the name of the profile that is added in Microsoft Outlook.

  For example:

  ```
  msiexec.exe /I Oracle CRM On Demand Desktop.5.0.0.x.exe OL_PROFILE="test PST"
  ```

Oracle CRM On Demand supports the following values for the OL_PROFILE parameter:

- **$DEFAULT.** Oracle CRM On Demand uses the default profile or default folder file. The default profile is the profile that you specify in Windows by choosing the Start menu, navigating to Control Panel, Mail, Show Profiles, and then specifying a profile in the window of the Always Use This Profile option. The default folder is the folder that is specified for email delivery in the OL_PROFILE parameter.
- **$PREFERRED.** The following conditions apply:
  - If you do not specify a value for the OL_PROFILE parameter, then Oracle CRM On Demand uses the value in $PREFERRED as the default value.
  - If you specify $PREFERRED as the value for the OL_PROFILE parameter, then Oracle CRM On Demand uses the same algorithm that it uses to determine the value of $DEFAULT. The exception is that if Oracle CRM On Demand does not find a default profile or finds a profile that is not valid, then Oracle CRM On Demand chooses any other suitable profile.

Upgrading Oracle CRM On Demand Desktop from Version 5.0 to Version 5.1

New versions of Oracle CRM On Demand Desktop are periodically made available to you through Oracle CRM On Demand. Upgrading your installation of Oracle CRM On Demand Desktop is similar to installing Oracle CRM On Demand Desktop. For more information on installing and hosting Oracle CRM On Demand Desktop, see *Process of Installing Oracle CRM On Demand Desktop*. To prevent data loss during an upgrade, complete the following procedure.

Note: It is recommended that you uninstall your existing version of Oracle CRM On Demand Desktop and install the latest version soon after it is made available.

To upgrade Oracle CRM On Demand Desktop from Version 5.0 to Version 5.1

1. Download the upgraded installer with the latest customization package from the Oracle Software Delivery Cloud Web site.

   For more information on downloading the Oracle CRM On Demand Desktop installer, see *Obtaining the Oracle CRM On Demand Desktop Installer*.
2. Update the files in the customization package for the current version of Oracle CRM On Demand Desktop.

   For more information on updating the files in the customization package, see *Updating Oracle CRM On Demand Desktop Product-Specific Files from Version 5.0 to Version 5.1*.
3. Customize Oracle CRM On Demand Desktop as required.

   For information on customizing Oracle CRM On Demand Desktop and the customization package, see *Customizing Oracle CRM On Demand Desktop*.
4. Schedule a date to perform the upgrade, and notify all Oracle CRM On Demand Desktop users.
5. Inform all Oracle CRM On Demand Desktop users that if they do not perform a synchronization before this date, then the previous version of Oracle CRM On Demand Desktop might not work.

   For more information, see *Preventing Data Loss When Upgrading a Customization Package*.
6. Add an end date to all currently active customization packages.
7. Create the new customization package administration records in Oracle CRM On Demand.

   The start date of this customization package must be equal to the scheduled upgrade date.
8. Upload the new installer to the required location.

   If the Oracle CRM On Demand Desktop user installs the new installer after the new customization package becomes active, then the new installer will work immediately. However, if the Oracle CRM On Demand Desktop user installs the new installer before the start date specified on the new customization package, then one of two possibilities can happen:

   - Oracle CRM On Demand Desktop might not function during the period leading up to the scheduled start date.
   - Oracle CRM On Demand Desktop might function but will require a synchronization on the scheduled start date.

Updating Oracle CRM On Demand Desktop Product-Specific Files from Version 5.0 to Version 5.1

If you have upgraded Oracle CRM On Demand Desktop from Version 5.0 to Version 5.1, then you must manually update all the product-specific files before you begin to use the current version of Oracle CRM On Demand Desktop. These updates preserve your configuration from Version 5.0 while maintaining compatibility with the metadata for the current release and allow you to use the new features.

In this procedure, you update the following files:

- forms.js
- business_logic.js
- helpers.js
- application_script.js
- connector_configuration.xml
- autoresolver.js
- dialogs.xml
- forms_12*.xml
- info.xml
- od_basic_mapping.xml
- od_meta_info.xml
- security_utils.js
- package_res.xml

For more information on customization package files, see *Files in the Customization Package*.

To update Oracle CRM On Demand Desktop product-specific files from Version 5.0 to Version 5.1

1. Update the forms.js file as follows:
    a. Locate the following code:

```
{
"caption": "obj_leads_plural",
"view_id": "leads:salesbook",
"allow_new": false,
"search_by": ["LeadFirstName",
"LeadLastName"
],
"image": "type_image:Lead:16",
"display_name_fn": function(ctx, item) {
 return item["LeadFirstName"] +" " + item["LeadLastName"];
},
```

    b. Replace this code with the following:

```
{
"caption": "obj_leads_plural",
"view_id": "leads:salesbook",
"allow_new": true,
"search_by": ["LeadFirstName", "LeadLastName"],
"image": "type_image:Lead:16",
"display_name_fn": function(ctx, item) {
 return item["LeadFirstName"] + " " + item["LeadLastName"];
},
```

c. Locate the following code:

```
if (ctx.item_ex.get_id() == null) {
 var created_through_salesbook = ctx.created_from_ctx != null &&
!!ctx.created_from_ctx.created_from_ctx_type;
 shared_on_open = helpers.get_defaults(ctx.session) != null &&
(created_through_salesbook || ctx.application
 .settings.get("SharedByDefault:NewItems") == 1);
 ctx.item_ex.set_property("SiebelShared", shared_on_open);
}
```

d. Replace this code with the following:

```
if (ctx.item_ex.get_id() == null) {
 var created_through_salesbook = ctx.created_from_ctx != null &&
!!ctx.created_from_ctx.created_from_ctx_type;
 shared_on_open = od_helpers.get_defaults(ctx.session) != null &&
(created_through_salesbook || ctx.application
 .settings.get("SharedByDefault:NewItems") == 1);
 ctx.item_ex.set_property("SiebelShared", shared_on_open);
}
```

e. Locate the following code:

```
form.share_contact.checked = true;
}
else
if (form.item.snapshot.ObjectState != 0 && ctx.ui.message_box(0,
ctx.session.res_string("msg_want_to_delete"),
ctx.session.res_string("msg_unshare_caption") + " " + ctx.item_ex.get_type(),
0x44) != 6) form.share_contact.checked = true;
}
else {
 if (!form.item.AddrConvertedFrom) {
 var cont_converter = new business_logic.contact_conversion_helper(ctx);
 cont_converter.convert_to_ondemand(ctx.item_ex,
cont_addr_opener.dialog());
iconcat_baddr(form.item);
}
 od_helpers.check_sharing_possibility(ctx, form.share_contact);
}
setShareBarCaption(ctx, form.share_contact, null, 1);
}
var form_saving = function()
```

f. Replace this code with the following:

```
form.share_contact.checked = true;
}
else
if (ctx.security_descriptor.is_synced() && ctx.ui.message_box(0,
ctx.session.res_string("msg_want_to_delete"),
ctx.session.res_string("msg_unshare_caption") + " " + ctx.item_ex.get_type(),
0x44) != 6) form.share_contact
.checked = true;
}
else {
 od_helpers.check_sharing_possibility(ctx, form.share_contact);
```

```
}
setShareBarCaption(ctx, form.share_contact, null, 1);
form.CompanyNameLbl.visible = !form.share_contact.checked;
form.CompanyName.visible = !form.share_contact.checked;
}
var form_saving = function()
```

g. Locate the following code:

```
//FORM EVENTS
form.on_saving.connect(form_saving);
form.on_saved.connect(form_saved);
}
// LEAD FORM SCRIPTS //
```

h. Replace this code with the following:

```
//FORM EVENTS
form.on_saving.connect(form_saving);
form.on_saved.connect(form_saved);
form.CompanyNameLbl.visible = !ctx.item_ex.get_property("SiebelShared");
form.CompanyName.visible = !ctx.item_ex.get_property("SiebelShared");
}

// LEAD FORM SCRIPTS //
```

i. Locate the following code:

```
var ownership_mode = ctx.security_descriptor.ownership_mode(),
 form = ctx.form,
 type = form.item.type_id,
 current_user_link = {
 "item_ex": ctx.item_ex,
 "with_id": helpers.get_current_user_id(ctx.session),
 "tag": "direct"
 },
 current_user_link_mvg = helpers.merge_contexts(current_user_link, {
 "tag": "mvg",
 "params": {
 "is_manual_operation": false
 }
 });
if (!is_activity_subform(type)) {
```

j. Replace this code with the following:

```
var ownership_mode = ctx.security_descriptor.ownership_mode(),
 form = ctx.form,
 type = form.item.type_id,
 current_user_link = {
 "item_ex": ctx.item_ex,
 "with_id": od_helpers.get_current_user_id(ctx.session),
 "tag": "direct"
 },
 current_user_link_mvg = helpers.merge_contexts(current_user_link, {
 "tag": "mvg",
 "params": {
 "is_manual_operation": false
 }
 });
if (!is_activity_subform(type)) {
```

k. Locate the following code:

```
if (ctrl.checked || shared) {
 caption = ctx.session.res_string("btn_shared");
} else {
 caption = helpers.replace_all("{full_name}",
ctx.form.item.snapshot.ContactFullName,
ctx.session.res_string("btn_share_contact"))
}
}
ctrl.caption = caption;
```

l. Replace this code with the following:

```
if (ctrl.checked || shared) {
 caption = ctx.session.res_string("btn_shared");
} else {
 var contact_name = ctx.form.item.get_instant("ContactFullName");
 if(!helpers.is_null_or_empty(contact_name)){
 caption = helpers.replace_all("{full_name}", contact_name,
ctx.session.res_string("btn_share_contact"));
 } else {
 caption = ctx.session.res_string("btn_share");
 }
}
}
ctrl.caption = caption;
```

2. Update the business_logic.js file as follows:

a. Locate the following code:

```
var result = new Array();
if (addr != "" && addr != null) {
 var curr_user_id = helpers.get_current_user_id(session);
 var filter = session.create_criteria("and");
 filter.add(session.create_expression("ContactEmail", "like", addr));
 filter.add(session.create_expression("SiebelShared", "eq", true));
```

b. Replace this code with the following:

```
var result = new Array();
if (addr != "" && addr != null) {
 var curr_user_id = od_helpers.get_current_user_id(session);
 var filter = session.create_criteria("and");
 filter.add(session.create_expression("ContactEmail", "like", addr));
 filter.add(session.create_expression("SiebelShared", "eq", true));
```

c. Locate the following code:

```
helpers.for_each(session.find_items("User", filter), function(item) {
 result.push(item.id)
});
if (result.length == 0 && account_addresses != null &&
include.helpers.contains(account_addresses, addr))
result.push(include.helpers.get_current_user_id(session));
}
return result;
}
```

d. Replace this code with the following:

```
helpers.for_each(session.find_items("User", filter), function(item) {
 result.push(item.id)
});
```

```
if (result.length == 0 && account_addresses != null &&
include.helpers.contains(account_addresses, addr))
result.push(include.od_helpers.get_current_user_id(session));
}
return result;
}
```

e. Locate the following code:

```
var result = new Array();
if (addr != "" && addr != null) {
 var curr_user_id = helpers.get_current_user_id(session);
 var filter = session.create_criteria("and");
 filter.add(session.create_expression("LeadEmail", "like", addr));
 helpers.for_each(session.find_items("Lead", filter), function(item) {
 result.push(item.id)
 });
```

f. Replace this code with the following:

```
var result = new Array();
if (addr != "" && addr != null) {
 var curr_user_id = od_helpers.get_current_user_id(session);
 var filter = session.create_criteria("and");
 filter.add(session.create_expression("LeadEmail", "like", addr));
 helpers.for_each(session.find_items("Lead", filter), function(item) {
 result.push(item.id)
 });
```

g. Locate the following code:

```
var mode = od_helpers.get_ownership_mode(ctx, action_ctx.type);
if (mode == "mixed" || mode == "owner") {
 var curr_user_id = helpers.get_current_user_id(ctx.session),
 with_user_id = action_ctx.with_id;
 var is_manual_operation = action_ctx.params == null ||
action_ctx.params.is_manual_operation;
 // prevent manual set primary to non-current user
```

h. Replace this code with the following:

```
var mode = od_helpers.get_ownership_mode(ctx, action_ctx.type);

if (mode == "mixed" || mode == "owner") {
 var curr_user_id = od_helpers.get_current_user_id(ctx.session),
 with_user_id = action_ctx.with_id;
 var is_manual_operation = action_ctx.params == null ||
action_ctx.params.is_manual_operation;
 // prevent manual set primary to non-current user
```

i. Locate the following code:

```
var item_ex = ctx.item_ex;
var default_type_id = item_ex ? item_ex.get_type() : null;
type_id = type_id || default_type_id;
var current_user_id = helpers.get_current_user_id(ctx.session);
if (current_user_id != null && type_id && item_ex) {
 var ownership_mode = od_helpers.get_ownership_mode(ctx, type_id);
```

j. Replace this code with the following:

```
var item_ex = ctx.item_ex;
var default_type_id = item_ex ? item_ex.get_type() : null;
type_id = type_id || default_type_id;
```

```
var current_user_id = od_helpers.get_current_user_id(ctx.session);
if (current_user_id != null && type_id && item_ex) {
 var ownership_mode = od_helpers.get_ownership_mode(ctx, type_id);
```

k. Locate the following code:

```
function prefill_owner_od(ctx) {
 var current_user_id = helpers.get_current_user_id(ctx.session);
 return current_user_id;
}
```

l. Replace this code with the following:

```
function prefill_owner_od(ctx){
 var current_user_id = od_helpers.get_current_user_id(ctx.session);
 return current_user_id;
}
```

m. Locate the following code:

```
}
if (!book_id) {
 var defaults = helpers.get_defaults(ctx.session),
 user_default_book_name = defaults != null ? defaults.DefaultBookName : "",
 user_default_book = ctx.session.find_item("Book",
od_helpers.create_expression_or_null_value_filter(ctx.session,"Label", "eq",
user_default_book_name));
 book_id = user_default_book != null ? user_default_book.Value : book_id;
```

n. Replace this code with the following:

```
}
if (!book_id) {
 var defaults = od_helpers.get_defaults(ctx.session),
 user_default_book_name = defaults != null ? defaults.DefaultBookName :"",
 user_default_book = ctx.session.find_item("Book",
od_helpers.create_expression_or_null_value_filter(ctx.session, "Label", "eq",
user_default_book_name));
 book_id = user_default_book != null ? user_default_book.Value :book_id;
```

o. Locate the following code:

```
var fill_owner = ownership_mode != "book" && !(ownership_mode == "mixed" &&
prefill_book(ctx, item_ex)); // prefill owner in mixed mode if no book was
prefilled
if (fill_owner) {
 current_user_id = helpers.get_current_user_id(ctx.session);
}
return current_user_id;
}
```

p. Replace this code with the following:

```
var fill_owner = ownership_mode != "book" && !(ownership_mode == "mixed" &&
prefill_book(ctx, item_ex)); // prefill owner in mixed mode if no book was
prefilled
if (fill_owner) {
 current_user_id = od_helpers.get_current_user_id(ctx.session);
}
return current_user_id;
}
```

q. Locate the following code:

```
function get_default_currency_code_od(ctx) {
 var defaults = helpers.get_defaults(ctx.session);
 var currency_code = null;
 if ((defaults != null) && (defaults["CurrencyCode"] != null)) {
```

r. Replace this code with the following:

```
function get_default_currency_code_od(ctx){
 var defaults = od_helpers.get_defaults(ctx.session);
 var currency_code = null;
 if ((defaults != null) && (defaults["CurrencyCode"] != null)) {
```

s. Locate the following code:

```
{
 var sales_process_id = null,
 filter = ctx.session.create_criteria("and"),
 defaults = helpers.get_defaults(ctx.session);
 if ((defaults != null) && (defaults["SalesProcessId"] != "")){
 // default sales process which is set for the user’s role
```

t. Replace this code with the following:

```
{
 var sales_process_id = null,
 filter = ctx.session.create_criteria("and"),
 defaults = od_helpers.get_defaults(ctx.session);
 if ((defaults != null) && (defaults["SalesProcessId"] != "")){
 // default sales process which is set for the user’s role
```

u. Locate the following code:

```
{
 var LanguageCode;
 var access_profile = ctx.session.res_string("lang_default_access_profile");
 var defaults = helpers.get_defaults(ctx.session);
 if ((defaults != null) && (defaults["LanguageCode"] != null)){
 // user LanguageCode
```

v. Replace this code with the following:

```
{
 var LanguageCode;
 var access_profile = ctx.session.res_string("lang_default_access_profile");
 var defaults = od_helpers.get_defaults(ctx.session);
 if ((defaults != null) && (defaults["LanguageCode"] != null)) {
 // user LanguageCode
```

w. Locate the following code:

```
function get_current_user_login(ctx) {
 var user_login = "";
 var defaults = include.helpers.get_defaults(ctx.session);
 if (defaults != null) {
 user_login = ctx.session.open_item(defaults.CurrentUser)["Login Name"];
```

x. Replace this code with the following:

```
function get_current_user_login(ctx) {
 var user_login = "";
 var defaults = include.od_helpers.get_defaults(ctx.session);
 if (defaults != null) {
```

```
 user_login = ctx.session.open_item(defaults.CurrentUser)["Login Name"];
```

y. Locate the following code:

```
scheme.objects.get_object("Activity").get_field("ObjectState")["initial_value"]
= 0;
scheme.objects.get_object("Activity").get_field("SiebelShared")["initial_value"
] = true;
scheme.objects.get_object("Activity").get_field("SuppressFileAs")["initial_valu
e_res"] ="lang_general_initial_fileas";
scheme.objects.get_object("Activity").get_field("Priority")["initial_value_res"
] = "lang_action_priority_low";
scheme.objects.get_object("Activity").get_field("DueDate")["initial_value_fn"]
= date_today_no_time;
scheme.objects.get_object("Activity")["initial_links_fn"] =
prefill_activity_owner;
scheme.objects.get_object("Activity").get_field("BookId")["initial_value_fn"] =
prefill_book;
```

z. Replace this code with the following:

```
scheme.objects.get_object("Activity").get_field("ObjectState")["initial_value"]
= 0;
 scheme.objects.get_object("Activity").get_field("SiebelShared")["initial_value"
] = true;
 scheme.objects.get_object("Activity").get_field("SuppressFileAs")["initial_valu
e_res"] = "lang_general_initial_fileas";
 scheme.objects.get_object("Activity").get_field("Priority")["initial_value_fn"]
= function(ctx, item_ex)
{
return get_pl_default_value(ctx, "ActivityPriorityPicklist",
"lang_action_priority_low");
}
scheme.objects.get_object("Activity").get_field("DueDate")["initial_value_fn"]
= date_today_no_time;
scheme.objects.get_object("Activity")["initial_links_fn"] =
prefill_activity_owner;
scheme.objects.get_object("Activity").get_field("BookId")["initial_value_fn"] =
prefill_book;
```

aa. Locate the following code:

```
scheme.objects.get_object("Contact").get_field("ObjectState")["initial_value"]
= 0;
scheme.objects.get_object("Contact")["initial_links_fn"] = prefill_owner;
scheme.objects.get_object("Contact").get_field("CurrencyCode")["initial_value_f
n"] = get_default_currency_code_od;
 scheme.objects.get_object("Contact").get_field("SuppressFileAs")["initial_value
_res"] = "lang_general_initial_fileas";
 scheme.objects.get_object("Contact").get_field("BookId")["initial_value_fn"] =
prefill_book;
 scheme.objects.get_object("Account").get_field("SuppressFileAs")["initial_value
_res"] = "lang_general_initial_fileas";
```

ab. Replace this code with the following:

```
scheme.objects.get_object("Contact").get_field("ObjectState")["initial_value"]
= 0;
scheme.objects.get_object("Contact")["initial_links_fn"] = prefill_owner;
scheme.objects.get_object("Contact").get_field("CurrencyCode")["initial_value_f
n"] =
 get_default_currency_code_od;
scheme.objects.get_object("Contact").get_field("SuppressFileAs")["initial_value
_fn"] = function(ctx, item_ex) {
 // prefill only if empty
 var existing_value = item_ex.fcd_item.get_instant("SuppressFileAs");
 var result = helpers.is_null_or_empty(existing_value) ?
ctx.session.res_string("lang_general_initial_fileas") : existing_value; return
result;
};
scheme.objects.get_object("Contact").get_field("BookId")["initial_value_fn"] =
prefill_book;
scheme.objects.get_object("Account").get_field("SuppressFileAs")["initial_value
_res"] = "lang_general_initial_fileas";
```

ac. Locate the following code:

```
// Helper functions
function has_defaults_and_online() {
 return helpers.get_defaults(ctx.session) != null &&
od_helpers.is_online(ctx);
}
function current_user_id() {
 return helpers.get_current_user_id(ctx.session);
}
function is_current_user(id)
```

ad. Replace this code with the following:

```
// Helper functions
function has_defaults_and_online() {
 return od_helpers.get_defaults(ctx.session) != null &&
od_helpers.is_online(ctx);
}
function current_user_id()
{
 return od_helpers.get_current_user_id(ctx.session);
}
function is_current_user(id)
```

ae. Locate the following code:

```
switch (ol.item_ex().get_type()) {
 case "Event":
 proxy.item_ex().set_property("Type",
ctx.session.res_string("lang_action_calendar_initial_type"));
 //proxy.item_ex().set_property("Status", "Unscheduled");
 proxy.item_ex().set_property("iCRMId",
ol.item_ex().get_property("SearchKey"));
 create_initial_associations(proxy, ol);
 break;
 case "Task":
 proxy.item_ex().set_property("Type",
ctx.session.res_string("lang_action_todo_initial_type"));
 proxy.item_ex().set_property("iCRMId",
ol.item_ex().get_property("SearchKey"));
 create_initial_associations(proxy, ol);
 break;
```

af. Replace this code with the following:

```
switch (ol.item_ex().get_type()){
 case "Event":
 proxy.item_ex().set_property("Type", get_pl_default_value(ctx,
"ActivityTypePicklist", "lang_action_calendar_initial_type"));
 //proxy.item_ex().set_property("Status", "Unscheduled");
 proxy.item_ex().set_property("iCRMId",
ol.item_ex().get_property("SearchKey"));
```

```
 create_initial_associations(proxy, ol);
 break;
 case "Task":
 proxy.item_ex().set_property("Type", get_pl_default_value(ctx,
"ActivityTypePicklist", "lang_action_todo_initial_type"));
 proxy.item_ex().set_property("iCRMId",
ol.item_ex().get_property("SearchKey"));
 create_initial_associations(proxy, ol);
 break;
```

ag. Locate the following code:

```
case "Mail":
//proxy.item_ex().set_property("Type", ol.item_ex().get_property("Inbound") ==
true ? "Email - Inbound" : "Email - Outbound");
 proxy.item_ex().set_property("Status",
ctx.session.res_string("lang_action_mail_initial_status"));
 proxy.item_ex().set_property("Type",
ctx.session.res_string("lang_action_mail_initial_type"));
 proxy.item_ex().set_property("SiebelShared",
ol.item_ex().get_property("Sent"));
 create_initial_associations(proxy, ol);
 break;
```

ah. Replace this code with the following:

```
case "Mail":
//proxy.item_ex().set_property("Type", ol.item_ex().get_property("Inbound") ==
true ? "Email - Inbound" : "Email - Outbound");
 proxy.item_ex().set_property("Status",
ctx.session.res_string("lang_action_mail_initial_status"));
 proxy.item_ex().set_property("Type", get_pl_default_value(ctx,
"ActivityTypePicklist", "lang_action_mail_initial_type"));
 proxy.item_ex().set_property("SiebelShared",
ol.item_ex().get_property("Sent"));
 create_initial_associations(proxy, ol);
 break;
```

ai. Locate the following code:

```
function contact_conversion_helper(ctx) {
 this.business_address = "b";
 this.home_address = "h";
 function null_to_empty(value) {
 return value == null ? "" : value;
 }
 var ol2od = {
 "US": "USA",
 "GB": "United Kingdom",
 "RU": "Russian Federation"
 }
 function verify_country(country, code) {
 if (null_to_empty(country) == "") return false;
 if (code != null && ol2od[code] != null) {
 return ol2od[code];
 }
 var od_country =
ctx.session.find_item("AccountPrimaryBillToCountryPicklist",
ctx.session.create_expression(
"Value", "eq", country));
 return od_country && od_country.Value;
 }
 function verify_state(state) {
 if (null_to_empty(state) == "") return false;
 var od_state = ctx.session.find_item("AccountPrimaryBillToStatePicklist",
ctx.session.create_expression("Value", "eq", state));
 return od_state && od_state.Value;

 }
 function convert_to_ondemand(item_ex, dialog) {
 //var item = ctx.form.item;
 if (item_ex.get_type() == "Contact") {
 var country = {};
 country[this.home_address] = item_ex.get_property("Home Address Country");
 country[this.business_address] = item_ex.get_property("Business Address
Country");
 var state = {};
 state[this.home_address] = item_ex.get_property("Home Address State");
 state[this.business_address] = item_ex.get_property("Business Address
State");
 var city = {};
 city[this.home_address] = item_ex.get_property("Home Address City");
 city[this.business_address] = item_ex.get_property("Business Address
City");
 var address = {};
 address[this.home_address] = item_ex.get_property("Home Address Street");
 address[this.business_address] = item_ex.get_property("Business Address
Street");
 var zip = {};
 zip[this.home_address] = item_ex.get_property("Home Address Postal Code");
 zip[this.business_address] = item_ex.get_property("Business Address Postal
Code");
 var country_code = {};
 country_code[this.home_address] = item_ex.get_property("Address Country
Code");
 country_code[this.business_address] = item_ex.get_property("Address
Country Code");
 var at = ""; // address_type_to_use
 var od_country
 if (od_country = verify_country(country.b, country_code.b)) at =
this.business_address;
 else if (od_country = verify_country(country.h, country_code.h)) at =
this.home_address;
 if (at != "" && dialog != null) {
 dialog.item = item_ex.fcd_item;
 item_ex.set_property("AddrConvertedFrom", at);
 dialog.country_value.value = od_country;
 if (od_country == "USA" && verify_state(state[at]))
dialog.state_value_pl.value = state[at];
 else dialog.state_value.value = state[at];
 dialog.city_value.value = city[at];
 dialog.street_value.value = address[at].replace(/\n/g, " ");
 dialog.zip_value.value = zip[at];
 dialog.save();
 }
 }
 }
 this.convert_to_ondemand = convert_to_ondemand;
 function empty_AddrConvertedFrom(item_ex) {
 if (item_ex.get_type() == "Contact") {
 //item_ex.set_property("AddrConvertedFrom", "");
 //item.save();
 }
 }
 function convert_to_outlook(item_ex) {
 if (item_ex.get_type() == "Contact") {
 var at = item_ex.get_property("AddrConvertedFrom") == "h" ? "Home " :
"Business ";
 var country = item_ex.get_property("AlternateCountry");
 var state = item_ex.get_property("AlternateStateProvince") ?
item_ex.get_property( "AlternateStateProvince") :
```

```
item_ex.get_property("AlternateProvince");
 var city = item_ex.get_property("AlternateCity");
 var address = item_ex.get_property("AlternateAddress1");
 var zip = item_ex.get_property("AlternateZipCode");
 if (null_to_empty(country) + null_to_empty(state) + null_to_empty(city) +
null_to_empty(address) + null_to_empty(zip) != "") {
 item_ex.set_property(at + "Address Country", country);
 item_ex.set_property(at + "Address State", state);
 item_ex.set_property(at + "Address City", city);
 item_ex.set_property(at + "Address Street", address);
 item_ex.set_property(at + "Address Postal Code", zip);
 //item.save();
 }
 }
 }
 this.convert_to_outlook = convert_to_outlook;
 this.converted_to_platform = function(ctx, options, item_ex) {
 empty_AddrConvertedFrom(item_ex);
 }
 this.converting_to_native = function(ctx, options, item_ex) {
 convert_to_outlook(item_ex);
 }
 this.set_visible = function(ctx, options, item_ex, visible) {
 if (!visible) item_ex.set_property("InvisibleOnGrids", visible ? null :
"true");
 }
}
```

aj. Replace this code with the following:

```
function contact_conversion_helper(ctx) {
 function null_to_empty(value) {
 return value == null ? "" : value;
 }
 function convert_to_outlook(ctx, item_ex) {
 if (item_ex.get_type() == "Contact") {
 var at = item_ex.get_property("AddrConvertedFrom");
 at = helpers.ci_contains(["Business", "Home", "Other"], at) ? at :
"Business";
 var country = item_ex.get_property("AlternateCountry");
 var state = item_ex.get_property("AlternateStateProvince") ?
item_ex.get_property( "AlternateStateProvince") :
item_ex.get_property("AlternateProvince");
 var city = item_ex.get_property("AlternateCity");
 var address = item_ex.get_property("AlternateAddress1");
 var zip = item_ex.get_property("AlternateZipCode");
 if (null_to_empty(country) + null_to_empty(state) + null_to_empty(city)
+ null_to_empty(address) + null_to_empty(zip) != "") {
 item_ex.set_property(at + " Address Country",
country_id_to_label(country));
 item_ex.set_property(at + " Address State", state);
 item_ex.set_property(at + " Address City", city);
 item_ex.set_property(at + " Address Street", address);
 item_ex.set_property(at + " Address Postal Code", zip);
 }
 }
 function country_id_to_label(country_id) {
 var result = null;
 if (!helpers.is_null_or_empty(country_id)) {
 var od_country =
ctx.session.find_item("AccountPrimaryBillToCountryPicklist",
ctx.session.create_expression( "Value", "eq", country_id));
 result = od_country ? od_country.Label : result;
 }
 return result;
 }
 }
 this.converted_to_platform = function(ctx, options, item_ex) {
 convert_ol_to_crm_address(ctx, item_ex);
 }
 this.converting_to_native = function(ctx, options, item_ex) {
 convert_to_outlook(ctx, item_ex);
 }
 this.set_visible = function(ctx, options, item_ex, visible) {
 if (!visible) item_ex.set_property("InvisibleOnGrids", visible ? null :
"true");
 }
 function convert_ol_to_crm_address(ctx, item_ex) {
 // copy native address to CRM fields
 var address_to_use;
 if (is_address_populated(item_ex, "Business")) {
 address_to_use = "Business";
 }else if (is_address_populated(item_ex, "Home")) {
 address_to_use = "Home";
 } else if (is_address_populated(item_ex, "Other")) {
 address_to_use = "Other";
 }
 if (address_to_use) {
 var crm_country_id = ol_to_crm_country(item_ex, address_to_use);
 item_ex.set_property("AlternateCountry", crm_country_id);
 if (crm_country_id == "USA") {
 var crm_state = ol_to_crm_state(item_ex, address_to_use);
 item_ex.set_property("AlternateStateProvince", crm_state);
 } else {
 item_ex.set_property("AlternateProvince",
item_ex.get_property(address_to_use + " Address State"));
 }
 item_ex.set_property("AlternateAddress1",
item_ex.get_property(address_to_use + " Address Street"));
 item_ex.set_property("AlternateCity",
item_ex.get_property(address_to_use + " Address City"));
 item_ex.set_property("AlternateZipCode",
item_ex.get_property(address_to_use + " Address Postal Code"));
 item_ex.set_property("AddrConvertedFrom", address_to_use);
 }
 function is_address_populated(item_ex, address_type) {
 var result = item_ex.get_property(address_type + " Address City") ||
item_ex.get_property(address_type + " Address Country") ||
item_ex.get_property(address_type + " Address Postal Code") ||
item_ex.get_property( address_type + " Address State") ||
item_ex.get_property(address_type + " Address Street");
 return !helpers.is_null_or_empty(result);
 }
 function ol_to_crm_country(item_ex, address_type) {
 var mapping = [{
 ol_code: 'AF',
 ol_name: 'Afghanistan',
 crm_id: 'Afghanistan'
 }, {
 ol_code: 'AL',
 ol_name: 'Albania',
 crm_id: 'Albania'
 }, {
 ol_code: 'DZ',
 ol_name: 'Algeria',
 crm_id: 'Algeria'
 }, {
 ol_code: 'AS',
 ol_name: 'American Samoa',
 crm_id: 'American Samoa'
 }, {
```

```
ol_code: 'AD',
ol_name: 'Andorra',
crm_id: 'Andorra'
}, {
ol_code: 'AO',
ol_name: 'Angola',
crm_id: 'Angola'
}, {
ol_code: 'AI',
ol_name: 'Anguilla',
crm_id: 'Anguilla'
}, {
ol_code: 'AG',
ol_name: 'Antigua and Barbuda',
crm_id: 'Antigua and Barbuda'
}, {
ol_code: 'AR',
ol_name: 'Argentina',
crm_id: 'Argentina'
}, {
ol_code: 'AM',
ol_name: 'Armenia',
crm_id: 'Armenia'
}, {
ol_code: 'AW',
ol_name: 'Aruba',
crm_id: 'Aruba'
},
//{ ol_code: '', ol_name: 'Ashmore and Cartier Islands', crm_id: ''},
{
ol_code: 'AU',
ol_name: 'Australia',
crm_id: 'Australia'
}, {
ol_code: 'AT',
ol_name: 'Austria',
crm_id: 'Austria'
}, {
ol_code: 'AZ',
ol_name: 'Azerbaijan',
crm_id: 'Azerbaijan'
}, {
ol_code: 'BS',
ol_name: 'Bahamas',
crm_id: 'Bahamas'
}, {
ol_code: 'BH',
ol_name: 'Bahrain',
crm_id: 'Bahrain'
}, {
ol_code: 'UM',
ol_name: 'Baker Island',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: 'BD',
ol_name: 'Bangladesh',
crm_id: 'Bangladesh'
}, {
ol_code: 'BB',
ol_name: 'Barbados',
crm_id: 'Barbados'
}, {
ol_code: 'BY',
ol_name: 'Belarus',
crm_id: 'Belarus'
}, {
```

```
ol_code: 'BE',
ol_name: 'Belgium',
crm_id: 'Belgium'
}, {
ol_code: 'BZ',
ol_name: 'Belize',
crm_id: 'Belize'
}, {
ol_code: 'BJ',
ol_name: 'Benin',
crm_id: 'Benin'
}, {
ol_code: 'BM',
ol_name: 'Bermuda',
crm_id: 'Bermuda'
}, {
ol_code: 'BT',
ol_name: 'Bhutan',
crm_id: 'Bhutan'
}, {
ol_code: 'BO',
ol_name: 'Bolivia',
crm_id: 'Bolivia'
}, {
ol_code: 'BQ',
ol_name: 'Bonaire',
crm_id: 'Bonaire, Sint Eustatius & Saba'
}, {
ol_code: 'BA',
ol_name: 'Bosnia and Herzegovina',
crm_id: 'Bosnia and Herzegovina'
}, {
ol_code: 'BW',
ol_name: 'Botswana',
crm_id: 'Botswana'
}, {
ol_code: 'BV',
ol_name: 'Bouvet Island',
crm_id: 'Bouvet Island'
}, {
ol_code: 'BR',
ol_name: 'Brazil',
crm_id: 'Brazil'
}, {
ol_code: 'IO',
ol_name: 'British Indian Ocean Territory',
crm_id: 'British Indian Ocean Territory'
}, {
ol_code: 'VG',
ol_name: 'British Virgin Islands',
crm_id: 'Virgin Islands (British)'
}, {
ol_code: 'BN',
ol_name: 'Brunei Darussalam',
crm_id: 'Brunei Darussalam'
}, {
ol_code: 'BG',
ol_name: 'Bulgaria',
crm_id: 'Bulgaria'
}, {
ol_code: 'BF',
ol_name: 'Burkina Faso',
crm_id: 'Burkina Faso'
}, {
ol_code: 'BI',
ol_name: 'Burundi',
crm_id: 'Burundi'
}, {
ol_code: 'KH',
ol_name: 'Cambodia',
crm_id: 'Cambodia'
}, {
ol_code: 'CM',
ol_name: 'Cameroon',
crm_id: 'Cameroon'
}, {
ol_code: 'CA',
ol_name: 'Canada',
crm_id: 'Canada'
}, {
ol_code: 'CV',
ol_name: 'Cape Verde',
crm_id: 'Cape Verde'
}, {
ol_code: 'KY',
ol_name: 'Cayman Islands',
crm_id: 'Cayman Islands'
}, {
ol_code: 'CF',
ol_name: 'Central African Republic',
crm_id: 'Central African Republic'
}, {
ol_code: 'TD',
ol_name: 'Chad',
crm_id: 'Chad'
}, {
ol_code: 'CL',
ol_name: 'Chile',
crm_id: 'Chile'
}, {
ol_code: 'CN',
ol_name: 'China',
crm_id: 'China'
}, {
ol_code: 'CX',
ol_name: 'Christmas Island',
crm_id: 'Christmas Island'
}, {
ol_code: 'CC',
ol_name: 'Cocos (Keeling) Islands',
crm_id: 'Cocos (Keeling) Islands'
}, {
ol_code: 'CO',
ol_name: 'Colombia',
crm_id: 'Colombia'
}, {
ol_code: 'KM',
ol_name: 'Comoros',
crm_id: 'Comoros'
}, {
ol_code: 'CG',
ol_name: 'Congo',
crm_id: 'Congo'
}, {
ol_code: 'CD',
ol_name: 'Congo (DRC)',
crm_id: 'Congo, Democratic Republic of'
}, {
ol_code: 'CK',
ol_name: 'Cook Islands',
crm_id: 'Cook Islands'
},
//{ ol_code: '', ol_name: 'Coral Sea Islands', crm_id: ''},
{
ol_code: 'CR',
ol_name: 'Costa Rica',
crm_id: 'Costa Rica'
}, {
ol_code: 'CI',
ol_name: 'Cote d\'Ivoire',
crm_id: 'Ivory Coast'
}, {
ol_code: 'HR',
ol_name: 'Croatia',
crm_id: 'Croatia'
}, {
ol_code: 'CU',
ol_name: 'Cuba',
crm_id: 'Cuba'
}, {
ol_code: 'CW',
ol_name: 'Curaçao',
crm_id: 'Curacao'
}, {
ol_code: 'CY',
ol_name: 'Cyprus',
crm_id: 'Cyprus'
}, {
ol_code: 'CZ',
ol_name: 'Czech Republic',
crm_id: 'Czech Republic'
}, {
ol_code: 'DK',
ol_name: 'Denmark',
crm_id: 'Denmark'
}, {
ol_code: 'DJ',
ol_name: 'Djibouti',
crm_id: 'Djibouti'
}, {
ol_code: 'DM',
ol_name: 'Dominica',
crm_id: 'Dominica'
}, {
ol_code: 'DO',
ol_name: 'Dominican Republic',
crm_id: 'Dominican Republic'
}, {
ol_code: 'EC',
ol_name: 'Ecuador',
crm_id: 'Ecuador'
}, {
ol_code: 'EG',
ol_name: 'Egypt',
crm_id: 'Egypt'
}, {
ol_code: 'SV',
ol_name: 'El Salvador',
crm_id: 'El Salvador'
}, {
ol_code: 'GQ',
ol_name: 'Equatorial Guinea',
crm_id: 'Equatorial Guinea'
}, {
ol_code: 'ER',
ol_name: 'Eritrea',
crm_id: 'Eritrea'
}, {
```

```
ol_code: 'EE',
ol_name: 'Estonia',
crm_id: 'Estonia'
}, {
ol_code: 'ET',
ol_name: 'Ethiopia',
crm_id: 'Ethiopia'
}, {
ol_code: 'FK',
ol_name: 'Falkland Islands (Islas Malvinas)',
crm_id: 'Falkland Islands (Malvinas)'
}, {
ol_code: 'FO',
ol_name: 'Faroe Islands',
crm_id: 'Faroe Islands'
}, {
ol_code: 'FJ',
ol_name: 'Fiji',
crm_id: 'Fiji'
}, {
ol_code: 'FI',
ol_name: 'Finland',
crm_id: 'Finland'
}, {
ol_code: 'FR',
ol_name: 'France',
crm_id: 'France'
}, {
ol_code: 'GF',
ol_name: 'French Guiana',
crm_id: 'French Guiana'
}, {
ol_code: 'PF',
ol_name: 'French Polynesia',
crm_id: 'French Polynesia'
}, {
ol_code: 'TF',
ol_name: 'French Southern and Antarctic Lands',
crm_id: 'French Southern Territories'
}, {
ol_code: 'GA',
ol_name: 'Gabon',
crm_id: 'Gabon'
}, {
ol_code: 'GM',
ol_name: 'Gambia',
crm_id: 'Gambia'
}, {
ol_code: 'GE',
ol_name: 'Georgia',
crm_id: 'Georgia'
}, {
ol_code: 'DE',
ol_name: 'Germany',
crm_id: 'Germany'
}, {
ol_code: 'GH',
ol_name: 'Ghana',
crm_id: 'Ghana'
}, {
ol_code: '',
ol_name: 'Gibraltar',
crm_id: 'Gibraltar'
}, {
ol_code: 'GR',
ol_name: 'Greece',
crm_id: 'Greece'
}, {
ol_code: 'GL',
ol_name: 'Greenland',
crm_id: 'Greenland'
}, {
ol_code: 'GD',
ol_name: 'Grenada',
crm_id: 'Grenada'
}, {
ol_code: 'GP',
ol_name: 'Guadeloupe',
crm_id: 'Guadeloupe'
}, {
ol_code: 'GU',
ol_name: 'Guam',
crm_id: 'Guam'
}, {
ol_code: 'GT',
ol_name: 'Guatemala',
crm_id: 'Guatemala'
}, {
ol_code: '',
ol_name: 'Guernsey',
crm_id: 'Guernsey'
}, {
ol_code: 'GN',
ol_name: 'Guinea',
crm_id: 'Guinea'
}, {
ol_code: 'GW',
ol_name: 'Guinea-Bissau',
crm_id: 'Guinea-bissau'
}, {
ol_code: 'GY',
ol_name: 'Guyana',
crm_id: 'Guyana'
}, {
ol_code: 'HT',
ol_name: 'Haiti',
crm_id: 'Haiti'
}, {
ol_code: 'HM',
ol_name: 'Heard Island and McDonald Islands',
crm_id: 'Heard and Mc Donald Islands'
}, {
ol_code: 'VA',
ol_name: 'Holy See (Vatican City)',
crm_id: 'Holy See (Vatican City State)'
}, {
ol_code: 'HN',
ol_name: 'Honduras',
crm_id: 'Honduras'
}, {
ol_code: 'HK',
ol_name: 'Hong Kong SAR',
crm_id: 'Hong Kong'
}, {
ol_code: 'UM',
ol_name: 'Howland Island',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: 'HU',
ol_name: 'Hungary',
crm_id: 'Hungary'
}, {
```

```
ol_code: 'IS',
ol_name: 'Iceland',
crm_id: 'Iceland'
}, {
ol_code: 'IN',
ol_name: 'India',
crm_id: 'India'
}, {
ol_code: 'ID',
ol_name: 'Indonesia',
crm_id: 'Indonesia'
}, {
ol_code: 'IR',
ol_name: 'Iran',
crm_id: 'Iran'
}, {
ol_code: 'IQ',
ol_name: 'Iraq',
crm_id: 'Iraq'
}, {
ol_code: 'IE',
ol_name: 'Ireland',
crm_id: 'Ireland'
}, {
ol_code: '',
ol_name: 'Isle of Man',
crm_id: 'Isle of Man'
}, {
ol_code: 'IL',
ol_name: 'Israel',
crm_id: 'Israel'
}, {
ol_code: 'IT',
ol_name: 'Italy',
crm_id: 'Italy'
}, {
ol_code: 'JM',
ol_name: 'Jamaica',
crm_id: 'Jamaica'
}, {
ol_code: 'JP',
ol_name: 'Japan',
crm_id: 'Japan'
}, {
ol_code: 'UM',
ol_name: 'Jarvis Island',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: '',
ol_name: 'Jersey',
crm_id: 'Jersey'
}, {
ol_code: 'UM',
ol_name: 'Johnston Atoll',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: 'JO',
ol_name: 'Jordan',
crm_id: 'Jordan'
}, {
ol_code: 'KZ',
ol_name: 'Kazakhstan',
crm_id: 'Kazakhstan'
}, {
ol_code: 'KE',
ol_name: 'Kenya',
crm_id: 'Kenya'
}, {
ol_code: 'UM',
ol_name: 'Kingman Reef',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: 'KI',
ol_name: 'Kiribati',
crm_id: 'Kiribati'
}, {
ol_code: 'KR',
ol_name: 'Korea',
crm_id: 'Korea'
}, {
ol_code: 'KP',
ol_name: 'North Korea',
crm_id: 'Korea, Democratic People\'s Rep'
}, {
ol_code: 'KW',
ol_name: 'Kuwait',
crm_id: 'Kuwait'
}, {
ol_code: 'KG',
ol_name: 'Kyrgyzstan',
crm_id: 'Kyrgyzstan'
}, {
ol_code: 'LA',
ol_name: 'Laos',
crm_id: 'Laos'
}, {
ol_code: 'LV',
ol_name: 'Latvia',
crm_id: 'Latvia'
}, {
ol_code: 'LB',
ol_name: 'Lebanon',
crm_id: 'Lebanon'
}, {
ol_code: 'LS',
ol_name: 'Lesotho',
crm_id: 'Lesotho'
}, {
ol_code: 'LR',
ol_name: 'Liberia',
crm_id: 'Liberia'
}, {
ol_code: 'LY',
ol_name: 'Libya',
crm_id: 'Libyan Arab Jamahiriya'
}, {
ol_code: 'LI',
ol_name: 'Liechtenstein',
crm_id: 'Liechtenstein'
}, {
ol_code: 'LT',
ol_name: 'Lithuania',
crm_id: 'Lithuania'
}, {
ol_code: 'LU',
ol_name: 'Luxembourg',
crm_id: 'Luxembourg'
}, {
ol_code: 'MO',
ol_name: 'Macau SAR',
crm_id: 'Macau'
}, {
```

```
ol_code: 'MK',
ol_name: 'Macedonia, FYRO',
crm_id: 'Macedonia'
}, {
ol_code: 'MG',
ol_name: 'Madagascar',
crm_id: 'Madagascar'
}, {
ol_code: 'MW',
ol_name: 'Malawi',
crm_id: 'Malawi'
}, {
ol_code: 'MY',
ol_name: 'Malaysia',
crm_id: 'Malaysia'
}, {
ol_code: 'MV',
ol_name: 'Maldives',
crm_id: 'Maldives'
}, {
ol_code: 'ML',
ol_name: 'Mali',
crm_id: 'Mali'
}, {
ol_code: 'MT',
ol_name: 'Malta',
crm_id: 'Malta'
}, {
ol_code: 'MH',
ol_name: 'Marshall Islands',
crm_id: 'Marshall Islands'
}, {
ol_code: 'MQ',
ol_name: 'Martinique',
crm_id: 'Martinique'
}, {
ol_code: 'MR',
ol_name: 'Mauritania',
crm_id: 'Mauritania'
}, {
ol_code: 'MU',
ol_name: 'Mauritius',
crm_id: 'Mauritius'
}, {
ol_code: 'YT',
ol_name: 'Mayotte',
crm_id: 'Mayotte'
}, {
ol_code: 'MX',
ol_name: 'Mexico',
crm_id: 'Mexico'
}, {
ol_code: 'FM',
ol_name: 'Micronesia',
crm_id: 'Micronesia, Federated States o'
}, {
ol_code: 'UM',
ol_name: 'Midway Islands',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: 'MD',
ol_name: 'Moldova',
crm_id: 'Moldova, Republic of'
}, {
ol_code: 'MC',
ol_name: 'Monaco',
crm_id: 'Monaco'
}, {
ol_code: 'MN',
ol_name: 'Mongolia',
crm_id: 'Mongolia'
}, {
ol_code: 'ME',
ol_name: 'Montenegro',
crm_id: 'Montenegro'
}, {
ol_code: 'MS',
ol_name: 'Montserrat',
crm_id: 'Montserrat'
}, {
ol_code: 'MA',
ol_name: 'Morocco',
crm_id: 'Morocco'
}, {
ol_code: 'MZ',
ol_name: 'Mozambique',
crm_id: 'Mozambique'
}, {
ol_code: 'MM',
ol_name: 'Myanmar',
crm_id: 'Myanmar'
}, {
ol_code: 'NA',
ol_name: 'Namibia',
crm_id: 'Namibia'
}, {
ol_code: 'NR',
ol_name: 'Nauru',
crm_id: 'Nauru'
}, {
ol_code: 'NP',
ol_name: 'Nepal',
crm_id: 'Nepal'
}, {
ol_code: 'NL',
ol_name: 'Netherlands',
crm_id: 'Netherlands'
}, {
ol_code: 'NC',
ol_name: 'New Caledonia',
crm_id: 'New Caledonia'
}, {
ol_code: 'NZ',
ol_name: 'New Zealand',
crm_id: 'New Zealand'
}, {
ol_code: 'NI',
ol_name: 'Nicaragua',
crm_id: 'Nicaragua'
}, {
ol_code: 'NE',
ol_name: 'Niger',
crm_id: 'Niger'
}, {
ol_code: 'NG',
ol_name: 'Nigeria',
crm_id: 'Nigeria'
}, {
ol_code: 'NU',
ol_name: 'Niue',
crm_id: 'Niue'
}, {
```

```
ol_code: 'NF',
ol_name: 'Norfolk Island',
crm_id: 'Norfolk Island'
}, {
ol_code: 'MP',
ol_name: 'Northern Mariana Islands',
crm_id: 'Northern Mariana Islands'
}, {
ol_code: 'NO',
ol_name: 'Norway',
crm_id: 'Norway'
}, {
ol_code: 'OM',
ol_name: 'Oman',
crm_id: 'Oman'
}, {
ol_code: 'PK',
ol_name: 'Pakistan',
crm_id: 'Pakistan'
}, {
ol_code: 'PW',
ol_name: 'Palau',
crm_id: 'Palau'
}, {
ol_code: 'PS',
ol_name: 'Palestinian Authority',
crm_id: 'Palestinian Authority'
}, {
ol_code: 'UM',
ol_name: 'Palmyra Atoll',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: 'PA',
ol_name: 'Panama',
crm_id: 'Panama'
}, {
ol_code: 'PG',
ol_name: 'Papua New Guinea',
crm_id: 'Papua New Guinea'
}, {
ol_code: 'PY',
ol_name: 'Paraguay',
crm_id: 'Paraguay'
}, {
ol_code: 'PE',
ol_name: 'Peru',
crm_id: 'Peru'
}, {
ol_code: 'PH',
ol_name: 'Philippines',
crm_id: 'Philippines'
}, {
ol_code: 'PN',
ol_name: 'Pitcairn Island',
crm_id: 'Pitcairn Islands'
}, {
ol_code: 'PL',
ol_name: 'Poland',
crm_id: 'Poland'
}, {
ol_code: 'PT',
ol_name: 'Portugal',
crm_id: 'Portugal'
}, {
ol_code: 'PR',
ol_name: 'Puerto Rico',
crm_id: 'Puerto Rico'
}, {
ol_code: 'QA',
ol_name: 'Qatar',
crm_id: 'Qatar'
}, {
ol_code: 'RE',
ol_name: 'Réunion',
crm_id: 'Reunion'
}, {
ol_code: 'RO',
ol_name: 'Romania',
crm_id: 'Romania'
}, {
ol_code: 'RU',
ol_name: 'Russia',
crm_id: 'Russian Federation'
}, {
ol_code: 'RW',
ol_name: 'Rwanda',
crm_id: 'Rwanda'
},
//{ ol_code: 'XS', ol_name: 'Saba', crm_id: ''},
{
ol_code: 'SH',
ol_name: 'Saint Helena',
crm_id: 'St. Helena'
}, {
ol_code: 'PM',
ol_name: 'Saint Pierre and Miquelon',
crm_id: 'St. Pierre and Miquelon'
}, {
ol_code: 'AS',
ol_name: 'Samoa',
crm_id: 'Samoa'
}, {
ol_code: 'SM',
ol_name: 'San Marino',
crm_id: 'San Marino'
}, {
ol_code: 'ST',
ol_name: 'São Tomé and Príncipe',
crm_id: 'Sao Tome and Principe'
}, {
ol_code: 'SA',
ol_name: 'Saudi Arabia',
crm_id: 'Saudi Arabia'
}, {
ol_code: 'SN',
ol_name: 'Senegal',
crm_id: 'Senegal'
}, {
ol_code: 'YU',
ol_name: 'Serbia',
crm_id: 'Serbia and Montenegro'
}, {
ol_code: 'SC',
ol_name: 'Seychelles',
crm_id: 'Seychelles'
}, {
ol_code: 'SL',
ol_name: 'Sierra Leone',
crm_id: 'Sierra Leone'
}, {
ol_code: 'SG', ol_name: 'Singapore',
crm_id: 'Singapore'
```

```
},
//{ ol_code: 'XE', ol_name: 'Sint Eustatius', crm_id: ''},
{
ol_code: 'SX',
ol_name: 'Sint Maarten',
crm_id: 'Sint Maarten'
}, {
ol_code: 'SK',
ol_name: 'Slovak Republic',
crm_id: 'Slovakia'
}, {
ol_code: 'SI',
ol_name: 'Slovenia',
crm_id: 'Slovenia'
}, {
ol_code: 'SB',
ol_name: 'Solomon Islands',
crm_id: 'Solomon Islands'
}, {
ol_code: 'SO',
ol_name: 'Somalia',
crm_id: 'Somalia'
}, {
ol_code: 'ZA',
ol_name: 'South Africa',
crm_id: 'South Africa'
}, {
ol_code: 'GS',
ol_name: 'South Georgia and the South Sandwich Islands',
crm_id: 'South Georgia and the South Sa'
}, {
ol_code: 'SS',
ol_name: 'South Sudan',
crm_id: 'South Sudan'
}, {
ol_code: 'ES',
ol_name: 'Spain',
crm_id: 'Spain'
}, {
ol_code: 'LK',
ol_name: 'Sri Lanka',
crm_id: 'Sri Lanka'
}, {
ol_code: 'KN',
ol_name: 'St. Kitts and Nevis',
crm_id: 'St. Kitts and Nevis'
}, {
ol_code: 'LC',
ol_name: 'St. Lucia',
crm_id: 'St. Lucia'
}, {
ol_code: 'VC',
ol_name: 'St. Vincent and the Grenadines',
crm_id: 'St. Vincent and Grenadines'
}, {
ol_code: 'SD',
ol_name: 'Sudan',
crm_id: 'Sudan'
}, {
ol_code: 'SR',
ol_name: 'Surinam',
crm_id: 'Suriname'
}, {
ol_code: 'SJ',
ol_name: 'Svalbard and Jan Mayen Islands',
crm_id: 'Svalbard and Jan Mayen Islands'
```

```
}, {
ol_code: 'SZ',
ol_name: 'Swaziland',
crm_id: 'Swaziland'
}, {
ol_code: 'SE',
ol_name: 'Sweden',
crm_id: 'Sweden'
}, {
ol_code: 'CH',
ol_name: 'Switzerland',
crm_id: 'Switzerland'
}, {
ol_code: 'SY',
ol_name: 'Syria',
crm_id: 'Syrian Arab Republic'
}, {
ol_code: 'TW',
ol_name: 'Taiwan',
crm_id: 'Taiwan'
}, {
ol_code: 'TJ',
ol_name: 'Tajikistan',
crm_id: 'Tajikistan'
}, {
ol_code: 'TZ',
ol_name: 'Tanzania',
crm_id: 'Tanzania, United Republic of'
}, {
ol_code: 'TH',
ol_name: 'Thailand',
crm_id: 'Thailand'
}, {
ol_code: 'TG',
ol_name: 'Togo',
crm_id: 'Togo'
}, {
ol_code: 'TK',
ol_name: 'Tokelau',
crm_id: 'Tokelau'
}, {
ol_code: 'TO',
ol_name: 'Tonga',
crm_id: 'Tonga'
}, {
ol_code: 'TT',
ol_name: 'Trinidad and Tobago',
crm_id: 'Trinidad and Tobago'
}, {
ol_code: 'TN',
ol_name: 'Tunisia',
crm_id: 'Tunisia'
}, {
ol_code: 'TR',
ol_name: 'Turkey',
crm_id: 'Turkey'
}, {
ol_code: 'TM',
ol_name: 'Turkmenistan',
crm_id: 'Turkmenistan'
}, {
ol_code: 'TC',
ol_name: 'Turks and Caicos Islands',
crm_id: 'Turks and Caicos Islands'
}, {
ol_code: 'TV',
ol_name: 'Tuvalu',
crm_id: 'Tuvalu'
}, {
ol_code: 'VI',
ol_name: 'U.S. Virgin Islands',
crm_id: 'Virgin Islands (U.S.)'
}, {
ol_code: 'US',
ol_name: 'United States of America',
crm_id: 'USA'
}, {
ol_code: 'UG',
ol_name: 'Uganda',
crm_id: 'Uganda'
}, {
ol_code: 'UA',
ol_name: 'Ukraine',
crm_id: 'Ukraine'
}, {
ol_code: 'AE',
ol_name: 'United Arab Emirates',
crm_id: 'United Arab Emirates'
}, {
ol_code: 'GB',
ol_name: 'United Kingdom',
crm_id: 'United Kingdom'
}, {
ol_code: 'UY',
ol_name: 'Uruguay',
crm_id: 'Uruguay'
}, {
ol_code: 'UZ',
ol_name: 'Uzbekistan',
crm_id: 'Uzbekistan'
}, {
ol_code: 'VU',
ol_name: 'Vanuatu',
crm_id: 'Vanuatu'
}, {
ol_code: 'VE',
ol_name: 'Venezuela',
crm_id: 'Venezuela'
}, {
ol_code: 'VN',
ol_name: 'Vietnam',
crm_id: 'Vietnam'
}, {
ol_code: 'UM',
ol_name: 'Wake Island',
crm_id: 'United States Minor Outlying I'
}, {
ol_code: 'WF',
ol_name: 'Wallis and Futuna',
crm_id: 'Wallis and Futuna Islands'
}, {
ol_code: 'YE',
ol_name: 'Yemen',
crm_id: 'Yemen'
}, {
ol_code: 'ZM',
ol_name: 'Zambia',
crm_id: 'Zambia'
}, {
ol_code: 'ZW',
ol_name: 'Zimbabwe',
crm_id: 'Zimbabwe'
```

```
 }
 ];
 var ol_country_text = item_ex.get_property(address_type + " Address
Country");
 var ol_country_text_lc = ol_country_text ? ol_country_text.toLowerCase() :
ol_country_text;
 var ol_country_code = item_ex.get_property(address_type + " Address Country
Code");
 var result = null;
 helpers.for_each2(mapping, function(el) {
 if (ol_country_code && ol_country_code == el.ol_code || ol_country_text
&& ol_country_text_lc ==
el.ol_code.toLowerCase() || ol_country_text_lc == el.ol_name.toLowerCase() ||
ol_country_text_lc == el.crm_id.toLowerCase()) {
result = el.crm_id;
 }
 });
 if (!result && !helpers.is_null_or_empty(ol_country_text)) {
var od_country = ctx.session.find_item("AccountPrimaryBillToCountryPicklist",
ctx.session.create_expression("Label", "eq", ol_country_text));
result = od_country ? od_country.Value : result;
 }
 return result;
 }
 function ol_to_crm_state(item_ex, address_type) {
 var ol_state = item_ex.get_property(address_type + " Address State");
 var result = null;
 if (ol_state) {
 var filter = ctx.session.create_criteria("or");
 filter.add(ctx.session.create_expression("Value", "eq", ol_state));
 filter.add(ctx.session.create_expression("Label", "eq", ol_state));
 var crm_state =
ctx.session.find_item("AccountPrimaryBillToStatePicklist", filter);
 result = crm_state ? crm_state.Value : result;
 }
 return result;
 }
 }
}
```

ak. Add the following code to the end of the file:

```
function get_pl_default_value(ctx, picklist_name, defailt_value_res)

{
 var filter = ctx.session.create_criteria("and"),default_value =
ctx.session.res_string(defailt_value_res);
 filter.add(ctx.session.create_expression("Value", "eq", default_value));
 var pl_item = ctx.session.find_item(picklist_name, filter);
 return pl_item ? default_value : null;
}
```

3. Copy the helpers.js file from the Oracle CRM On Demand Desktop 5.1 package and add it to your customization package.
4. Update the application_script.js file as follows:

   a. Locate the following code:

```
// Control panel
var cp_pages = [{
 native: false,
 layout: "cp_record_set_page",
 available: true,
 image: "type_image:Book:32",
 caption: application.session.res_string("lbl_record_set_caption"),
 description: application.session.res_string("record_set_tab_description")
}, {
 native: true,
 layout: "cp_filters_page",
 available: true
}, {
 native: true,
 layout: "cp_issues_page",
 available: true
}, {
 native: true,
 layout: "cp_conflicts_page",
 available: true
}, {
 native: true,
 layout: "cp_duplicates_page",
 available: true
}, {
 native: true,
 layout: "cp_confirmations_page",
 available: true
}];
```

b. Replace this code with the following:

```
// Control panel
var cp_pages = [{
 native: false,
 layout: "cp_record_set_page",
 available: true,
 image: "type_image:Book:32",
 caption: application.session.res_string("lbl_record_set_caption"),
 description: application.session.res_string("record_set_tab_description")
}, {
 native: true,
 layout: "cp_filters_page",
 available: true
}];
```

c. Locate the following code:

```
if (ol_id != null) od_helpers.open_remote(g_ctx,
application.session.open_item(ol_id));
}
else {
 g_ctx.data_model.triggers.call(g_ctx, {
 "link_to": type,
 "with_id": oid,
 "operation": "show"
```

d. Replace this code with the following:

```
if(ol_id != null) od_helpers.open_remote(g_ctx,
application.session.open_item(ol_id));
}
else {
 g_ctx.data_model.triggers.call(g_ctx_for_cp, {
 "link_to": type,
 "with_id": oid,
 "operation": "show"
```

e. Add the following code to the end of the file:

```
var g_ctx_for_cp = {custom_data: {}}; // required to open object on dialog from CP
helpers.merge_contexts(g_ctx, g_ctx_for_cp);
```

5. Update the connector_configuration.xml file as follows:

   a. Locate the following code:

```
<value>0</value>
</preset>
</sliding_dates_presets>
<features enable_sync_exclusions="true" >
 <bindings>
 <type name="Opportunity">
 <binding>
```

   b. Replace this code with the following:

```
<value>0</value>
</preset>
</sliding_dates_presets>
<features enable_sync_exclusions="true" autorevert_illegal_deletions="true">
 <bindings>
 <type name="Opportunity">
 <binding>
```

6. Update the autoresolver.js file as follows:

   a. Locate the following code:

```
var remote_set = conflict.remote_values();
var local_set = conflict.local_values();
if(local_set != null && local_set.get_field("CanAutoresolveToRemoteDelete") &&
remote_set == null) conflict.resolve_remote();
if (remote_set == null || local_set == null) return; // update-delete
```

   b. Replace this code with the following:

```
var remote_set = conflict.remote_values();
var local_set = conflict.local_values();
if(local_set != null && local_set.get_field("CanAutoresolveToRemoteDelete") &&
remote_set == null){
 conflict.resolve_remote();
 return;
}
if (remote_set == null || local_set == null) return; // update-delete
```

7. Update the dialogs.xml file as follows:

   a. Locate the following code:

```
<stack layout="vert" padding="10" spacing="10">
 <cell size="80">
 <stack layout="horz" spacing="5">
 <cell size="200">
 <stack layout="vert" padding="5" spacing="5">
 <cell size="20">
```

   b. Replace this code with the following:

```
<stack layout="vert" padding="10" spacing="10">
 <cell size="80">
 <stack layout="horz" spacing="5">
 <cell size="300">
 <stack layout="vert" padding="5" spacing="5">
 <cell size="20">
 <static id="criteria"
```

8. Update the forms_12*.xml file as follows:

   a. Locate the following code:

   ```
   </cell>
   </stack>
   </cell>
   <cell size="211">
    <stack layout="horz">
    <cell size="7">
    <stack layout="vert" spacing="1">
   ```

   b. Replace this code with the following:

   ```
   </cell>
   </stack>
   </cell>
   <cell size="238">
    <stack layout="horz">
    <cell size="7">
    <stack layout="vert" spacing="1">
   ```

   c. Locate the following code:

   ```
   <control id="lbl_file_as" window_id="0x11a4" tab_order="7"/> </cell>
   <cell size="22">
    <static id="0x20010" tab_order="9"> <text>#lbl_time_zone</text> </static>
   </cell>
   <cell size="22">
    <static id="0x20012" tab_order="11"> <text>#lbl_contact_currency</text> </
   static>
   </cell>
   <cell size="22">
    <static id="0x20014" tab_order="13"> <text>#lbl_contact_type</text> </static>
   </cell>
   <cell size="22">
    <static id="0x20016" tab_order="15"> <text>#lbl_lead_source</text> </static>
   </cell>
   ```

   d. Replace this code with the following:

   ```
   <control id="lbl_file_as" window_id="0x11a4" tab_order="7"/> </cell>
   <cell size="22">
    <control id="CompanyNameLbl" window_id="0x11a2" tab_order="9"/> </cell>
   <cell size="22">
    <static id="0x20010" tab_order="11"> <text>#lbl_time_zone</text> </static>
   </cell>
   <cell size="22">
    <static id="0x20012" tab_order="13"> <text>#lbl_contact_currency</text> </
   static>
   </cell>
   <cell size="22">
    <static id="0x20014" tab_order="15"> <text>#lbl_contact_type</text> </static>
   </cell>
   <cell size="22">
    <static id="0x20016" tab_order="17"> <text>#lbl_lead_source</text> </static>
   </cell>
   ```

   e. Locate the following code:

```
<control id="cb_file_as" window_id="0x1182" tab_order="8"/> </cell>
<cell size="22">
 <edit id="TimeZoneName" max_chars="100" tab_order="10">
 <field value="string">TimeZoneName</field>
 </edit>
</cell>
<cell size="22">
 <combobox id="CurrencyCode" tab_order="12">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="AccountCurrencyCodePicklist"/>
 <order_by>
```

f. Replace this code with the following:

```
<control id="cb_file_as" window_id="0x1182" tab_order="8"/> </cell>
<cell size="22">
 <control id="CompanyName" window_id="0x1181" tab_order="10"/> </cell>
<cell size="22">
 <edit id="TimeZoneName" max_chars="100" tab_order="12">
 <field value="string">TimeZoneName</field>
 </edit>
</cell>
<cell size="22">
 <combobox id="CurrencyCode" tab_order="14">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="AccountCurrencyCodePicklist"/>
 <order_by>
```

g. Locate the following code:

```
</combobox>
</cell>
<cell size="22">
 <combobox id="ContactType" tab_order="14">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="ContactContactTypePicklist"/>
 <order_by>
```

h. Replace this code with the following:

```
</combobox>
</cell>
<cell size="22">
 <combobox id="ContactType" tab_order="16">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="ContactContactTypePicklist"/>
 <order_by>
```

i. Locate the following code:

```
</combobox>
</cell>
<cell size="22">
 <combobox id="LeadSource" tab_order="16">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="ContactLeadSourcePicklist"/>
 <order_by>
```

j. Replace this code with the following:

```
</combobox>
</cell>
<cell size="22">
```

```
 <combobox id="LeadSource" tab_order="18">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="ContactLeadSourcePicklist"/>
 <order_by>
```

k. Locate the following code:

```
<cell size="16">
 <stack layout="horz">
 <cell size="110">
 <static id="0x20050" tab_order="40"> <text>#head_description</text> </
static>
 </cell>
```

l. Replace this code with the following:

```
<cell size="16">
 <stack layout="horz">
 <cell size="110">
 <static id="0x20050" tab_order="240"> <text>#head_description</text> </
static>
 </cell>
```

m. Locate the following code:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <control id="description" tab_order="41" window_id="0x103f"/> </cell>
 </stack>
</cell>
```

n. Replace this code with the following:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <control id="description" tab_order="241" window_id="0x103f"/>
 </cell>
 </stack>
</cell>
```

o. Locate the following code:

```
<cell size="16">
 <stack layout="horz">
 <cell size="70">
 <static id="0x20043" tab_order="42"> <text>#head_activities</text> </
static>
 </cell>
```

p. Replace this code with the following:

```
<cell size="16">
 <stack layout="horz">
 <cell size="70">
 <static id="0x20043" tab_order="242"> <text>#head_activities</text> </
static>
 </cell>
```

q. Locate the following code:

```
<cell>
```

```
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="activities_view" tab_order="43">
 <source type="auto" name="ActivityJointContact"/>
 <view id="activities:form_view-mvg"/>
 <restriction>
```

r. Replace this code with the following:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="activities_view" tab_order="243">
 <source type="auto" name="ActivityJointContact"/>
 <view id="activities:form_view-mvg"/>
 <restriction>
```

s. Locate the following code:

```
<cell size="22">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="140" attraction="far"> <button id="btn_delete_activity"
tab_order="46" image="button_image:Remove:16" align="left">
<text>#lbl_remove_activity</text>
</button> </cell>
 <cell size="140" attraction="far"> <button id="btn_new_task"
tab_order="45" image="type_image:Activity:16" align="left">
<text>#btn_new_task</text>
</button> </cell>
 <cell size="140" attraction="far">
<button id="btn_new_appt" tab_order="44" image="type_image:Activity:16"
align="left">
<text>#btn_new_appointment</text>
</button>
</cell>
```

t. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="140" attraction="far"> <button id="btn_delete_activity"
tab_order="246" image="button_image:Remove:16" align="left">
<text>#lbl_remove_activity</text>
</button> </cell>
 <cell size="140" attraction="far">
<button id="btn_new_task" tab_order="245" image="type_image:Activity:16"
align="left">
<text>#btn_new_task</text>
</button> </cell>
 <cell size="140" attraction="far">
<button id="btn_new_appt" tab_order="244" image="type_image:Activity:16"
align="left">
<text>#btn_new_appointment</text>
</button>
</cell>
```

u. Locate the following code:

```
<cell size="16">
 <stack layout="horz">
 <cell size="35">
 <static id="0x20052" tab_order="47"> <text>#head_notes</text> </static>
 </cell>
```

v. Replace this code with the following:

```
<cell size="16">
 <stack layout="horz">
 <cell size="35">
 <static id="0x20052" tab_order="247"> <text>#head_notes</text> </
static>
 </cell>
```

w. Locate the following code:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="notes_view" tab_order="48">
 <source type="auto" name="Contact.Note"/>
 <view id="crm_notes:form_view"/>
 <restriction>
```

x. Replace this code with the following:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="notes_view" tab_order="248">
 <source type="auto" name="Contact.Note"/>
 <view id="crm_notes:form_view"/>
 <restriction>
```

y. Locate the following code:

```
<cell size="22">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="140" attraction="far"> <button id="btn_delete_note"
tab_order="50" image="button_image:Remove:16" align="left">
<text>#lbl_delete_note</text>
</button></cell>
 <cell size="140" attraction="far"> <button id="btn_add_note" tab_order="49"
image="type_image:Note:16" align="left">
<text>#lbl_new_note</text>
</button>
</cell>
```

z. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="140" attraction="far"> <button id="btn_delete_note"
tab_order="250" image="button_image:Remove:16" align="left">
<text>#lbl_delete_note</text>
</button> </cell>
 <cell size="140" attraction="far"> <button id="btn_add_note" tab_order="249"
image="type_image:Note:16" align="left">
<text>#lbl_new_note</text>
</button>
</cell>
```

aa. Locate the following code:

```
<cell>
 <stack layout="horz">
 <cell size="85">
 <static id="PhoneNumbersCaption" tab_order="17">
```

```
<text>#head_phone_pumbers</text> </static>
 </cell>
```

ab. Replace this code with the following:

```
<cell>
 <stack layout="horz">
 <cell size="85">
 <static id="PhoneNumbersCaption" tab_order="19">
<text>#head_phone_pumbers</text> </static>
 </cell>
```

ac. Locate the following code:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_phone1" tab_order="18" window_id="0x1113"/> </cell>
 <cell size="23">
 <control id="dd_phone1" tab_order="19" window_id="0x110a"/> </cell>
 </stack>
</cell>
```

ad. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_phone1" tab_order="20" window_id="0x1113"/> </cell>
 <cell size="23">
 <control id="dd_phone1" tab_order="21" window_id="0x110a"/> </cell>
 </stack>
</cell>
```

ae. Locate the following code:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_phone4" tab_order="21" window_id="0x1116"/> </cell>
 <cell size="23">
 <control id="dd_phone4" tab_order="22" window_id="0x110d"/> </cell>
 </stack>
</cell>
```

af. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_phone4" tab_order="23" window_id="0x1116"/></cell>
 <cell size="23">
 <control id="dd_phone4" tab_order="24" window_id="0x110d"/> </cell>
 </stack>
</cell>
```

ag. Locate the following code:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_phone3" tab_order="24" window_id="0x1115"/> </cell>
 <cell size="23">
 <control id="dd_phone3" tab_order="25" window_id="0x110c"/> </cell>
 </stack>
</cell>
```

ah. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_phone3" tab_order="26" window_id="0x1115"/> </cell>
 <cell size="23">
 <control id="dd_phone3" tab_order="27" window_id="0x110c"/> </cell>
 </stack>
</cell>
```

ai. Locate the following code:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_email" tab_order="27" window_id="0x1101"/> </cell>
 <!--<cell size="5" attraction="far"></cell>-->
 <cell size="23" attraction="far">
 <control id="dd_email" tab_order="28" window_id="0x111b"/> </cell>
 </stack>
</cell>
```

aj. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz">
 <cell size="138">
 <control id="btn_email" tab_order="29" window_id="0x1101"/> </cell>
 <!--<cell size="5" attraction="far"></cell>-->
 <cell size="23" attraction="far">
 <control id="dd_email" tab_order="30" window_id="0x111b"/> </cell>
 </stack>
</cell>
```

ak. Locate the following code:

```
<cell>
 <stack spacing="5" layout="vert">
 <cell size="22">
 <control id="edit_phone1" tab_order="20" window_id="0x1001"/> </cell>
 <cell size="22">
 <control id="edit_phone4" tab_order="23" window_id="0x1004"/> </cell>
 <cell size="22">
 <control id="edit_phone3" tab_order="26" window_id="0x1003"/> </cell>
 <cell size="22">
 <control id="edit_email" tab_order="29" window_id="0x1018"/> </cell>
 <cell size="22">
 <checkbox id="0x20015" tab_order="30">
 <field>NeverEmail</field> <text>#lbl_never_email</text> </checkbox>
```

al. Replace this code with the following:

```
<cell>
 <stack spacing="5" layout="vert">
 <cell size="22">
 <control id="edit_phone1" tab_order="22" window_id="0x1001"/></cell>
 <cell size="22">
 <control id="edit_phone4" tab_order="25" window_id="0x1004"/> </cell>
 <cell size="22">
 <control id="edit_phone3" tab_order="28" window_id="0x1003"/> </cell>
 <cell size="22">
 <control id="edit_email" tab_order="31" window_id="0x1018"/> </cell>
 <cell size="22">
 <checkbox id="0x20015" tab_order="32">
 <field>NeverEmail</field> <text>#lbl_never_email</text> </checkbox>
```

am. Locate the following code:

```
<cell size="161">
 <stack spacing="5" layout="vert">
 <cell size="22"> <button id="btn_account_address" tab_order="31">
<text>#btn_account_address_caption</text>
</button> </cell>
 <cell size="22"> <button id="btn_contact_address" tab_order="33">
<text>#btn_contact_address_caption</text>
</button> </cell>
```

an. Replace this code with the following:

```
<cell size="161">
 <stack spacing="5" layout="vert">
 <cell size="22"> <button id="btn_account_address" tab_order="231">
<text>#btn_account_address_caption</text>
</button> </cell>
 <cell size="22"> <button id="btn_contact_address" tab_order="233">
<text>#btn_contact_address_caption</text>
</button> </cell>
```

ao. Locate the following code:

```
<cell>
 <stack spacing="5" layout="vert">
 <cell size="22">

 <scriptable_edit id="BusinessAddressString" tab_order="32">
 <field value="string"/> </scriptable_edit>
 </cell>
```

ap. Replace this code with the following:

```
<cell>
 <stack spacing="5" layout="vert">
 <cell size="22">
 <scriptable_edit id="BusinessAddressString" tab_order="232">
 <field value="string"/> </scriptable_edit>
 </cell>
```

aq. Locate the following code:

```
<cell size="22">
 <stack layout="horz">
 <cell>
 <scriptable_edit id="PrimaryAddrString" tab_order="34"> <field
value="string"/>
 </scriptable_edit>
 </cell>
```

ar. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz">
 <cell>
 <scriptable_edit id="PrimaryAddrString" tab_order="234"> <field
value="string"/>
```

```
 </scriptable_edit>
 </cell>
```

as. Locate the following code:

```
<cell size="161">
 <stack spacing="5" layout="vert">
 <cell size="22">
 <static id="0x20008" tab_order="35"> <text>#lbl_contact_team</text> </
static>
 </cell>
```

at. Replace this code with the following:

```
<cell size="161">
 <stack spacing="5" layout="vert">
 <cell size="22">
 <static id="0x20008" tab_order="235"> <text>#lbl_contact_team</text> </
static>
 </cell>
```

au. Locate the following code:

```
<cell size="22">
 <stack layout="horz">
 <cell>
 <autocomplete_list id="TeamToContact" tab_order="36">
 <items format=":[:{UserId@User/(:[:(FirstName):])}:]
:[:{UserId@User/(:[:(LastName):])}:]">
 <source type="auto" name="Contact.User.Association"/>
 <restriction>
```

av. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz">
 <cell>
 <autocomplete_list id="TeamToContact" tab_order="236">
 <items format=":[:{UserId@User/(:[:(FirstName):])}:]
:[:{UserId@User/(:[:(LastName):])}:]">
 <source type="auto" name="Contact.User.Association"/>
 <restriction>
```

aw. Locate the following code:

```
</cell>
<cell size="5"/>
<cell size="22">
<button id="btn_mvgContactTeam" tab_order="37">
<text>...</text>
</button>
</cell>
```

ax. Replace this code with the following:

```
</cell>
<cell size="5"/>
<cell size="22">
<button id="btn_mvgContactTeam" tab_order="237">
<text>...</text>
</button>
</cell>
```

ay. Locate the following code:

```
<cell size="22">
 <stack layout="horz">
 <cell>
 <scriptable_edit id="book_of_business" tab_order="38">
 <field value="string"/>
 </scriptable_edit>
 </cell>
 <cell size="5"/>
 <cell size="22"> <button id="btn_book_select" tab_order="39">
<text>#btn_book_select</text>
</button> </cell>
```

az. Replace this code with the following:

```
<cell size="22">
 <stack layout="horz">
 <cell>
 <scriptable_edit id="book_of_business" tab_order="238">
 <field value="string"/>
 </scriptable_edit>
 </cell>
 <cell size="5"/>
 <cell size="22"> <button id="btn_book_select" tab_order="239">
<text>#btn_book_select</text>
</button> </cell>
```

ba. Locate the following code:

```
<cell size="16">
 <stack layout="horz">
 <cell size="95">
 <static id="0x20046" tab_order="51"> <text>#head_opportunities</text>
</static>
 </cell>
```

bb. Replace this code with the following:

```
<cell size="16">
 <stack layout="horz">
 <cell size="95">
 <static id="0x20046" tab_order="251"> <text>#head_opportunities</text>
</static>
 </cell>
```

bc. Locate the following code:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="opportunities_view" tab_order="52">
 <source type="auto" name="OpportunityJointContact"/>
 <view id="opportunities:mvg"/>
 <restriction>
```

bd. Replace this code with the following:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="opportunities_view" tab_order="252">
 <source type="auto" name="OpportunityJointContact"/>
 <view id="opportunities:mvg"/>
 <restriction>
```

be. Locate the following code:

```
<cell size="22" attraction="far">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="160" attraction="far"> <button id="btn_delete_opportunity"
tab_order="54" image="button_image:Remove:16" align="left">
<text>#lbl_remove_opportunity</text>
</button> </cell>
 <cell size="160" attraction="far"> <button id="btn_mvgOpportunity"
tab_order="53" image="type_image:Opportunity:16" align="left">
<text>#lbl_new_opportunity</text>
</button> </cell>
```

bf. Replace this code with the following:

```
<cell size="22" attraction="far">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="160" attraction="far"> <button id="btn_delete_opportunity"
tab_order="254" image="button_image:Remove:16" align="left">
<text>#lbl_remove_opportunity</text>
</button> </cell>
 <cell size="160" attraction="far"> <button id="btn_mvgOpportunity"
tab_order="253" image="type_image:Opportunity:16" align="left">
<text>#lbl_new_opportunity</text>
</button> </cell>
```

bg. Locate the following code:

```
<cell size="16">
 <stack layout="horz">
 <cell size="110">
 <static id="head_leads" tab_order="55"> <text>#head_leads</text> </
static>
 </cell>
```

bh. Replace this code with the following:

```
<cell size="16">
 <stack layout="horz">
 <cell size="110">
 <static id="head_leads" tab_order="255"> <text>#head_leads</text> </
static>
 </cell>
```

bi. Locate the following code:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="leads_view" tab_order="56">
 <source type="auto" name="Lead"/>
 <view id="leads:form_view"/>
 <restriction>
```

bj. Replace this code with the following:

```
<cell>
 <stack layout="horz" padding="5">
 <cell>
 <data_view id="leads_view" tab_order="256">
 <source type="auto" name="Lead"/>
 <view id="leads:form_view"/>
 <restriction>
```

bk. Locate the following code:

```
<cell size="21" attraction="far">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="160" attraction="far"> <button id="btn_delete_lead"
tab_order="58" image="button_image:Remove:16" align="left">
<text>#lbl_delete_lead</text>
</button> </cell>
 <cell size="160" attraction="far"> <button id="btn_new_lead"
tab_order="57" image="type_image:Lead:16" align="left">
<text>#lbl_new_lead</text>
</button> </cell>
```

bl. Replace this code with the following:

```
<cell size="21" attraction="far">
 <stack layout="horz" spacing="5" padding="5">
 <cell size="160" attraction="far"> <button id="btn_delete_lead"
tab_order="258" image="button_image:Remove:16" align="left">
<text>#lbl_delete_lead</text>
</button> </cell>
 <cell size="160" attraction="far"> <button id="btn_new_lead"
tab_order="257" image="type_image:Lead:16" align="left">
<text>#lbl_new_lead</text>
</button> </cell>
```

bm. Locate the following code:

```
<control id="lbl Job title" window_id="0x11a3"/> </cell>
<cell>
 <control id="lbl Company" window_id="0x11a2"/> </cell>
<cell>
 <control id="edit Company" window_id="0x1181"/> </cell>
<cell>
 <control id="Image" window_id="0x1108"/> </cell>
<cell>
```

bn. Replace this code with the following:

```
<control id="lbl Job title" window_id="0x11a3"/> </cell>
<cell>
 <control id="Image" window_id="0x1108"/> </cell>
<cell>
```

bo. Locate the following code:

```
<stack layout="vert" padding="0" spacing="0">
 <cell>
 <combobox id="priority" tab_order="9">
 <items format=":[:(Label):]" value_column="Value"
has_null_item="false">
 <source type="auto" name="ActivityPriorityPicklist"/>
 <order_by>
 <order ascend="true">SortOrder</order>
```

bp. Replace this code with the following:

```
<stack layout="vert" padding="0" spacing="0">
 <cell>
 <combobox id="priority" tab_order="9">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="ActivityPriorityPicklist"/>
 <order_by>
 <order ascend="true">SortOrder</order>
```

bq. Locate the following code:

```
</cell>
<cell size="21">
 <combobox id="Type" tab_order="25">
 <items format=":[:(Label):]" value_column="Value" has_null_item="false">
 <source type="auto" name="ActivityTypePicklist"/>
 <order_by>
 <order ascend="true">SortOrder</order>
```

br. Replace this code with the following:

```
</cell>
 <cell size="21">
 <combobox id="Type" tab_order="25">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="ActivityTypePicklist"/>
 <order_by>
 <order ascend="true">SortOrder</order>
```

bs. Locate the following code:

```
</cell>
 <cell size="21">
 <combobox id="Type" tab_order="25">
 <items format=":[:(Label):]" value_column="Value"
has_null_item="false">
 <source type="auto" name="ActivityTypePicklist"/>
 <order_by>
 <order ascend="true">SortOrder</order>
```

bt. Replace this code with the following:

```
</cell>
 <cell size="21">
 <combobox id="Type" tab_order="25">
 <items format=":[:(Label):]" value_column="Value" has_null_item="true">
 <source type="auto" name="ActivityTypePicklist"/>
 <order_by>
 <order ascend="true">SortOrder</order>
```

9. Update the info.xml as follows:

a. Locate the following code:

```
<package_info xmlns="http://invisiblecrm.com/platform/info">
 <product>
 <name>Oracle CRM On Demand Desktop</name>
 <version>5.0.0.7</version>
 <server_version>21</server_version>
 </product>
 <compatibility preferred_product="5.0.0.7">
 <products>5.0.0.1-5.0.0.99</products>
 <schemas>5.0.0.1-5.0.0.99</schemas>
 </compatibility></package_info>
```

b. Replace this code with the following:

```
<package_info xmlns="http://invisiblecrm.com/platform/info">
 <product>
 <name>Oracle CRM On Demand Desktop</name>
 <version>5.1.0.13</version>
 <server_version>21</server_version>
 </product>
 <compatibility preferred_product="5.1.0.13">
 <products>5.1.0.1-5.1.0.99</products>
 <schemas>5.1.0.1-5.1.0.99</schemas>
 </compatibility>
</package_info>
```

Note: You must also perform this update to the forms_12.xml file and any localized files, such as forms_12.ja_JP.xml, forms_12.ko_KR.xml, forms_12.zh_CN.xml, forms_12.zh_HK.xml, forms_12.zh_MO.xml, forms_12.zh_SG.xml, and forms_12.zh_TW.xml.

10. Update the od_basic_mapping.xml as follows:

   a. Locate the following code:

```
</writer>
</field>
<field id="AccountId" ver="3">
 <reader>
 <mapi_user>
 <user_field id="od AccountId" ol_field_type="1"></user_field>
```

   b. Replace this code with the following:

```
</writer>
</field>
<field id="AccountId" ver="4">
 <reader>
 <mapi_user>
 <user_field id="od AccountId" ol_field_type="1"></user_field>
```

   c. Locate the following code:

```
</outlook_user>
<link_fields>
 <field from="AccountName" to="AccountName"></field>
 <field from="AccountName" to="SuppressFileAs"></field>
 <field from="Location" to="AccountLocation"></field>
 <field from="PrimaryBillToCountry" to="PrimaryCountry"></field>
 <field from="PrimaryBillToStreetAddress" to="PrimaryAddress"></field>
```

   d. Replace this code with the following:

```
</outlook_user>
<link_fields>
 <field from="AccountName" to="AccountName"></field>
 <field from="Location" to="AccountLocation"></field>
 <field from="PrimaryBillToCountry" to="PrimaryCountry"></field>
 <field from="PrimaryBillToStreetAddress" to="PrimaryAddress"></field>
```

   e. Locate the following code:

```
</outlook_std>
</writer>
</field>
<field id="Business Address City">
 <reader>
 <outlook_std>
```

   f. Replace this code with the following:

```
</outlook_std>
</writer>
</field>
<field id="Home Address Country Code">
 <reader>
 <mapi_named>
 <name_id id="0x80DA" ps="{00062004-0000-0000-C000-000000000046}"></
name_id>
 <convertor>
 <string/> </convertor>
 </mapi_named>
 </reader>
</field>
<field id="Business Address City">
<reader>
<outlook_std>
```

g. Locate the following code:

```
</outlook_std>
</writer>
</field>
<!-- Service fields-->
<field id="Address Country Code">
```

h. Replace this code with the following:

```
</outlook_std>
</writer>
</field>
<field id="Business Address Country Code">
 <reader>
 <mapi_named>
 <name_id id="0x80DB" ps="{00062004-0000-0000-C000-000000000046}"></
name_id>
 <convertor>
 <string/></convertor>
 </mapi_named>
 </reader>
</field>
<field id="Other Address City" ver="2">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A5F0000"></mapi_tag>
 <convertor>
 <string/></convertor>
 </mapi_std>
 </reader>
 <writer>
 <mapi_std>
 <mapi_tag id="0x3A5F0000"></mapi_tag>
 <convertor>
 <string/> </convertor>
 </mapi_std>
 </writer>
</field>
<field id="Other Address Country">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A600000"></mapi_tag>
 <convertor>
 <string/> </convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="OtherAddressCountry"></outlook_field>
 <convertor>
 <string/> </convertor>
 </outlook_std>
 </writer>
</field>
<field id="Other Address Postal Code">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A610000"></mapi_tag>
 <convertor>
 <string/> </convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="OtherAddressPostalCode"></outlook_field>
 <convertor>
 <string/> </convertor>
 </outlook_std>
 </writer>
</field>
<field id="Other Address State">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A620000"></mapi_tag>
 <convertor>
 <string/> </convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="OtherAddressState"></outlook_field>
 <convertor>
 <string/> </convertor>
 </outlook_std>
 </writer>
</field>
<field id="Other Address Street">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A630000"></mapi_tag>
 <convertor>
 <string/> </convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="OtherAddressStreet"></outlook_field>
 <convertor>
 <multiline_string/> </convertor>
 </outlook_std>
 </writer>
</field>
<field id="Other Address Country Code">
 <reader>
 <mapi_named>
 <name_id id="0x80DC" ps="{00062004-0000-0000-C000-000000000046}"></
name_id>
 <convertor>
 <string/> </convertor>
 </mapi_named>
 </reader>
</field>
```

```
<!-- Service fields-->
<field id="Address Country Code">
```

11. Update the od_meta_info.xml as follows:

    a. Locate the following code:

    ```
    <field Name="LastAssignmentSubmissionDate" Label="Last Assignment Submission
    Date" DataType="DTYPE_TEXT" IsFilterable='no' IsHidden="yes" CRMName="Last
    Assignment Submission Date" />
    <field Name="LastCallDate" Label="Last Call Date" DataType="DTYPE_TEXT"
    IsFilterable='no' IsHidden="yes" CRMName="Last Call Date" />
    <field Name="Location" Label="Location" DataType="DTYPE_TEXT" IsFilterable='yes'
    />
    <field Name="MainFax" Label="Main Fax" DataType="DTYPE_TEXT" IsFilterable='no'
    CRMName="Main Fax Number" />
    <field Name="MainPhone" Label="Main Phone" DataType="DTYPE_TEXT"
    IsFilterable='yes' CRMName="Main Phone Number" />
    <field Name="ManagedPartnerFlag" Label="Managed Partner Flag"
    DataType="DTYPE_BOOL" IsFilterable='no' IsHidden="yes" CRMName="Managed Partner
    Flag" />
    <field Name="MarketPotential" Label="Market Potential" DataType="DTYPE_TEXT"
    IsFilterable='no' IsHidden="yes" CRMName="Market Potential" />
    <field Name="MarketSegment" Label="Market Segment" DataType="DTYPE_TEXT"
    IsFilterable='no' IsHidden="yes" CRMName="Primary Market" />
    ```

    b. Replace this code with the following:

    ```
    <field Name="LastAssignmentSubmissionDate" Label="Last Assignment Submission
    Date" DataType="DTYPE_TEXT" IsFilterable='no' IsHidden="yes" CRMName="Last
    Assignment Submission Date" />
    <field Name="LastCallDate" Label="Last Call Date" DataType="DTYPE_TEXT"
    IsFilterable='no' IsHidden="yes" CRMName="Last Call Date" />
    <field Name="Location" Label="Location" DataType="DTYPE_TEXT" IsFilterable='yes'
    />
    <field Name="MainFax" Label="Main Fax" DataType="DTYPE_PHONE" IsFilterable='no'
    CRMName="Main Fax Number" />
    <field Name="MainPhone" Label="Main Phone" DataType="DTYPE_PHONE"
    IsFilterable='yes' CRMName="Main Phone Number" />
    <field Name="ManagedPartnerFlag" Label="Managed Partner Flag"
    DataType="DTYPE_BOOL" IsFilterable='no' IsHidden="yes" CRMName="Managed Partner
    Flag" />
    <field Name="MarketPotential" Label="Market Potential" DataType="DTYPE_TEXT"
    IsFilterable='no' IsHidden="yes" CRMName="Market Potential" />
    <field Name="MarketSegment" Label="Market Segment" DataType="DTYPE_TEXT"
    IsFilterable='no' IsHidden="yes" CRMName="Primary Market" />
    ```

    c. Locate the following code:

    ```
    <field Name="BookName" Label="BookName" DataType="DTYPE_TEXT" CRMName="Primary
    Position Name" >
    <suppress_on Upstream="true" />
    </field>
    <field Name="CellularPhone" Label="Cellular Phone" DataType="DTYPE_TEXT"
    CRMName="Cellular Phone #" />
    <field Name="ClientStatus" Label="ClientStatus" DataType="DTYPE_TEXT"
    IsFilterable="true" HasPicklist="true">
    <suppress_on Upstream="true" Get="true" />
    </field>
    ```

    d. Replace this code with the following:

    ```
    <field Name="BookName" Label="BookName" DataType="DTYPE_TEXT" CRMName="Primary
    Position Name" >
    ```

```
<suppress_on Upstream="true" />
</field>
<field Name="CellularPhone" Label="Cellular Phone" DataType="DTYPE_PHONE"
CRMName="Cellular Phone #" />
<field Name="ClientStatus" Label="ClientStatus" DataType="DTYPE_TEXT"
IsFilterable="true" HasPicklist="true">
<suppress_on Upstream="true" Get="true" />
</field>
```

e. Locate the following code:

```
<field Name="Department" Label="Department" DataType="DTYPE_TEXT"
IsFilterable='no' IsHidden="yes" />
<field Name="Description" Label="Description" DataType="DTYPE_TEXT"
CRMName="Comment" />
<field Name="Gender" Label="Gender" DataType="DTYPE_TEXT" IsFilterable='no'
IsHidden="yes" CRMName="M/F" />
<field Name="HomePhone" Label="Home Phone" DataType="DTYPE_TEXT"
IsFilterable='no' CRMName="Home Phone #" />
<field Name="Id" Label="Id" DataType="DTYPE_ID" IsPrimaryKey="yes"
IsFilterable='no' />
<field Name="JobTitle" Label="Job Title" DataType="DTYPE_TEXT" CRMName="Job
Title" />
<field Name="LeadSource" Label="Lead Source" DataType="DTYPE_TEXT"
HasPicklist="yes" IsFilterable='no' CRMName="Lead Source" />
```

f. Replace this code with the following:

```
<field Name="Department" Label="Department" DataType="DTYPE_TEXT"
IsFilterable='no' IsHidden="yes" />
<field Name="Description" Label="Description" DataType="DTYPE_TEXT"
CRMName="Comment" />
<field Name="Gender" Label="Gender" DataType="DTYPE_TEXT" IsFilterable='no'
IsHidden="yes" CRMName="M/F" />
<field Name="HomePhone" Label="Home Phone" DataType="DTYPE_PHONE"
IsFilterable='no' CRMName="Home Phone #" />
<field Name="Id" Label="Id" DataType="DTYPE_ID" IsPrimaryKey="yes"
IsFilterable='no' />
<field Name="JobTitle" Label="Job Title" DataType="DTYPE_TEXT" CRMName="Job
Title" />
<field Name="LeadSource" Label="Lead Source" DataType="DTYPE_TEXT"
HasPicklist="yes" IsFilterable='no' CRMName="Lead Source" />
```

g. Locate the following code:

```
<field Name="Territory" Label="Territory" DataType="DTYPE_TEXT"
IsFilterable='no' CRMName="Occam Territory" />
<field Name="TimeZoneName" Label="Time Zone Name" DataType="DTYPE_TEXT"
IsFilterable='no' CRMName="Time Zone Name - Translation" />
<field Name="TimeZoneId" Label="Time Zone Id" DataType="DTYPE_ID"
IsFilterable='no' IsHidden="yes" />
<field Name="WorkFax" Label="Work Fax" DataType="DTYPE_TEXT" IsFilterable='no'
CRMName="Fax Phone #" />
<field Name="WorkPhone" Label="Work Phone" DataType="DTYPE_TEXT" CRMName="Work
Phone #" />
</object>
<object TypeId="Contact.Address" Label="Address" LabelPlural="Address"
ViewMode="Sales Rep" IntObjName="Contact" XmlElemName="Address"
XmlCollectionElemName="ListOfAddress" PrimFlgFldname="IsPrimary"
EnableGetIDsBatching="yes" >
```

h. Replace this code with the following:

```
<field Name="Territory" Label="Territory" DataType="DTYPE_TEXT"
```

```
IsFilterable='no' CRMName="Occam Territory" />
<field Name="TimeZoneName" Label="Time Zone Name" DataType="DTYPE_TEXT"
IsFilterable='no' CRMName="Time Zone Name - Translation" />
<field Name="TimeZoneId" Label="Time Zone Id" DataType="DTYPE_ID"
IsFilterable='no' IsHidden="yes" />
<field Name="WorkFax" Label="Work Fax" DataType="DTYPE_PHONE" IsFilterable='no'
CRMName="Fax Phone #" />
<field Name="WorkPhone" Label="Work Phone" DataType="DTYPE_PHONE" CRMName="Work
Phone #" />
</object>
<object TypeId="Contact.Address" Label="Address" LabelPlural="Address"
ViewMode="Sales Rep" IntObjName="Contact" XmlElemName="Address"
XmlCollectionElemName="ListOfAddress" PrimFlgFldname="IsPrimary"
EnableGetIDsBatching="yes" >
```

i. Locate the following code:

```
<field Name='CampaignExternalSystemId' Label='Campaign External System Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" CRMName="Campaign
External Id" />
<field Name='CampaignId' Label='Campaign Id' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" CRMName="Campaign Id" />
<field Name='CampaignIntegrationId' Label='Campaign Integration Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" CRMName="Campaign
Integration Id" />
<field Name='CellularPhone' Label='Cellular Phone' DataType='DTYPE_TEXT'
CRMName="Cellular Phone #" />
<field Name='City' Label='City' DataType='DTYPE_TEXT' />
<field Name='Company' Label='Company' DataType='DTYPE_TEXT' IsFilterable='no'
CRMName="Company Name" />
<field Name='ContactExternalSystemId' Label='Contact External System Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" CRMName="Contact External
Id" />
```

j. Replace this code with the following:

```
<field Name='CampaignExternalSystemId' Label='Campaign External System Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" CRMName="Campaign
External Id" />
<field Name='CampaignId' Label='Campaign Id' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" CRMName="Campaign Id" />
<field Name='CampaignIntegrationId' Label='Campaign Integration Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" CRMName="Campaign
Integration Id" />
<field Name='CellularPhone' Label='Cellular Phone' DataType='DTYPE_PHONE'
CRMName="Cellular Phone #" />
<field Name='City' Label='City' DataType='DTYPE_TEXT' />
<field Name='Company' Label='Company' DataType='DTYPE_TEXT' IsFilterable='no'
CRMName="Company Name" />
<field Name='ContactExternalSystemId' Label='Contact External System Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" CRMName="Contact External
Id" />
```

k. Locate the following code:

```
<field Name='Description' Label='Description' DataType='DTYPE_TEXT'
IsFilterable='no' />
<field Name='EstimatedCloseDate' Label='Estimated Close Date'
DataType='DTYPE_UTCDATETIME' CRMName="Estimated Close Date" />
<field Name='ExternalSystemId' Label='External System Id' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" CRMName="External System Id" />
<field Name='FaxPhone' Label='Fax Phone' DataType='DTYPE_TEXT' IsFilterable='no'
CRMName="Fax Phone #" />
<field Name='FinancialAccountBalance' Label='Financial Account Balance'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
```

```
<field Name='FinancialAccountCurrencyCode' Label='Financial Account Currency
Code' DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
<field Name='FinancialAccountExchangeDate' Label='Financial Account Exchange
Date' DataType='DTYPE_DATE' IsFilterable='no' IsHidden="yes" />
```

l. Replace this code with the following:

```
<field Name='Description' Label='Description' DataType='DTYPE_TEXT'
IsFilterable='no' />
<field Name='EstimatedCloseDate' Label='Estimated Close Date'
DataType='DTYPE_UTCDATETIME' CRMName="Estimated Close Date" />
<field Name='ExternalSystemId' Label='External System Id' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" CRMName="External System Id" />
<field Name='FaxPhone' Label='Fax Phone' DataType='DTYPE_PHONE'
IsFilterable='no' CRMName="Fax Phone #" />
<field Name='FinancialAccountBalance' Label='Financial Account Balance'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
<field Name='FinancialAccountCurrencyCode' Label='Financial Account Currency
Code' DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
<field Name='FinancialAccountExchangeDate' Label='Financial Account Exchange
Date' DataType='DTYPE_DATE' IsFilterable='no' IsHidden="yes" />
```

m. Locate the following code:

```
<field Name='PolicyTotalPremium' Label='Policy Total Premium'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
<field Name='PolicyType' Label='Policy Type' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" />
<field Name='PotentialRevenue' Label='Potential Revenue' DataType='DTYPE_NUMBER'
IsFilterable='no' CRMName="Potential Revenue" />
<field Name='PrimaryPhone' Label='Primary Phone' DataType='DTYPE_TEXT'
CRMName="Work Phone #" />
<field Name='PrincipalPartnerChannelAccountManagerAlias'
Label='PrincipalPartnerChannelAccountManagerAlias' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" />
<field Name='PrincipalPartnerExternalSystemId' Label='Principal Partner External
System Id' DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
<field Name='PrincipalPartnerId' Label='Principal Partner Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
```

n. Replace this code with the following:

```
<field Name='PolicyTotalPremium' Label='Policy Total Premium'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
<field Name='PolicyType' Label='Policy Type' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" />
<field Name='PotentialRevenue' Label='Potential Revenue' DataType='DTYPE_NUMBER'
IsFilterable='no' CRMName="Potential Revenue" />
<field Name='PrimaryPhone' Label='Primary Phone' DataType='DTYPE_PHONE'
CRMName="Work Phone #" />
<field Name='PrincipalPartnerChannelAccountManagerAlias'
Label='PrincipalPartnerChannelAccountManagerAlias' DataType='DTYPE_TEXT'
IsFilterable='no' IsHidden="yes" />
<field Name='PrincipalPartnerExternalSystemId' Label='Principal Partner External
System Id' DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
<field Name='PrincipalPartnerId' Label='Principal Partner Id'
DataType='DTYPE_TEXT' IsFilterable='no' IsHidden="yes" />
```

12. Update the security_utils.js file as follows:

a. Locate the following code:

```
}
options = helpers.merge_contexts(default_options, options);
```

```
var MODIFICATION_BIT = 0x00000001;
var DELETION_BIT = 0x00000001;
var SYNC_BIT = 0x20000000;
var PINNED_BIT = 0x08000000;
options.temp_cache = {};
```

b. Replace this code with the following:

```
}
options = helpers.merge_contexts(default_options, options);
var PINNED_BIT = 0x08000000;
options.temp_cache = {};
```

c. Locate the following code:

```
this.is_synced = function(){
 if (options.temp_cache.is_sync_value == null)
options.temp_cache.is_sync_value = ((this.object_state() & SYNC_BIT) != 0);
 return options.temp_cache.is_sync_value;
}
this.is_pinned = function() {
 if (options.temp_cache.is_pinned_value == null)
options.temp_cache.is_pinned_value = ((this.object_state() & PINNED_BIT) != 0);
 return options.temp_cache.is_pinned_value;
}
this.current_user_is_primary = function()
{
```

d. Replace this code with the following:

```
this.is_synced = function() {
 if (options.temp_cache.is_sync_value == null)
options.temp_cache.is_sync_value = ((this.object_state() &
helpers.object_states.synchronized) === helpers.object_states.synchronized);
 return options.temp_cache.is_sync_value;
}
this.is_pinned = function() {
 if (options.temp_cache.is_pinned_value == null)
options.temp_cache.is_pinned_value = ((this.object_state() & PINNED_BIT) ===
PINNED_BIT);
 return options.temp_cache.is_pinned_value;
}
this.is_indirectly_visible = function()
{
 if (options.temp_cache.is_indirectly_visible_value == null)
options.temp_cache.is_indirectly_visible_value = ((this.object_state() &
helpers.object_states.indirectly_visible) ===
helpers.object_states.indirectly_visible);
 return options.temp_cache.is_indirectly_visible_value;
}
this.current_user_is_primary = function()
{
```

e. Locate the following code:

```
this.delete_access = function() {
 var can_delete = this.current_user_is_primary() && this.ownership_mode() !=
"book" || !this.is_synced();
 var value = new result_struct(can_delete);
 options.delete_filter(ctx, item_ex, value);
 return value.get_result();
```

f. Replace this code with the following:

```
this.delete_access = function(){
 var can_delete = !this.is_indirectly_visible() && (
this.current_user_is_primary() && this.ownership_mode() != "book" ||
!this.is_synced() );
 var value = new result_struct(can_delete);
 options.delete_filter(ctx, item_ex, value);
 return value.get_result();
```

g. Locate the following code:

```
this.modify_access = function() {
 var value = new result_struct(this.ownership_mode() != "owner" ||
this.current_user_is_in_team() || item_ex.get_type() ==
"Opportunity.ProductRevenue" || this.delete_access());
 options.modify_filter(ctx, item_ex, value);
 return value.get_result();
}
```

h. Replace this code with the following:

```
this.modify_access = function() {
 var can_modify = !this.is_indirectly_visible() && ( this.ownership_mode() !=
"owner" || this.current_user_is_in_team() || item_ex.get_type() ==
"Opportunity.ProductRevenue" || this.delete_access() );
 var value = new result_struct(can_modify);
 options.modify_filter(ctx, item_ex, value);
 return value.get_result();
}
```

i. Locate the following code:

```
var employee_options = {
 "link_to": "User",
 "get_current": function(ctx) { return
helpers.get_current_user_id(ctx.session);
 }
}
function item_unshared(ctx, item_ex, value)
```

j. Replace this code with the following:

```
 var employee_options = {
 "link_to": "User",
 "get_current": function(ctx) { return
od_helpers.get_current_user_id(ctx.session);
 }
}
function item_unshared(ctx, item_ex, value)
```

k. Locate the following code:

```
base_security.call(this, ctx, item_ex, helpers.merge_contexts(employee_options,
{}));
this.delete_access = function() {
 var can_delete = helpers.ids_equal(ctx.session,
item_ex.get_property("OwnerId"), helpers.get_current_user_id(ctx.session)) &&
this.ownership_mode() != "book" || !this.is_synced();
 return item_ex.get_id() == null || can_delete;
}
this.modify_access = function() {
 return this.ownership_mode() != "owner" || this.delete_access() ||
helpers.ids_equal(ctx.session, item_ex.get_property("SalesRepId"),
helpers.get_current_user_id(ctx.session));
```

```
}
this.link_access = function(link) {
```

l. Replace this code with the following:

```
base_security.call(this, ctx, item_ex, helpers.merge_contexts(employee_options,
{}));
this.delete_access = function() {
 var can_delete = !this.is_indirectly_visible() &&
(helpers.ids_equal(ctx.session, item_ex.get_property("OwnerId"),
od_helpers.get_current_user_id(ctx.session)) && this.ownership_mode() != "book"
|| ! this.is_synced());
 return item_ex.get_id() == null || can_delete;
}
this.modify_access = function() {
 return !this.is_indirectly_visible() && (this.ownership_mode() != "owner" ||
this.delete_access() ||
helpers.ids_equal(ctx.session, item_ex.get_property("SalesRepId"),
od_helpers.get_current_user_id(ctx.session)));
}
this.link_access = function(link) {
```

13. Update the package_res.xml file as follows:

a. Locate the following code:

```
<str key="msg_book_must_be_custom">Record must be associated to a custom book.</
str>
<str key="msg_owner_must_be_empty">Owner must be empty.</str>
<str key="inaccessible_book">&lt;Inaccessible book&gt;</str>
<str key="lbl_include_book_subitems">Include Sub-Items</str>
<str key="book_selector_top_node">$obj_book_plural$</str>
<str key="msg_ownership_log" skip_translation="yes">Ownership mode of {0} is
{1}</str>
<str key="msg_Specifying_both_Book_and_Owner_is_prohibited">Specifying both Book
and Owner is prohibited.</str>
```

b. Replace this code with the following:

```
<str key="msg_book_must_be_custom">Record must be associated to a custom book.</
str>
<str key="msg_owner_must_be_empty">Owner must be empty.</str>
<str key="inaccessible_book">&lt;Inaccessible book&gt;</str>
<str key="lbl_include_book_subitems">Include Sub-Books</str>
<str key="book_selector_top_node">$obj_book_plural$</str>
<str key="msg_ownership_log" skip_translation="yes">Ownership mode of {0} is
{1}</str>
<str key="msg_Specifying_both_Book_and_Owner_is_prohibited">Specifying both Book
and Owner is prohibited.</str>
```

c. Locate the following code:

```
<str key="msg_book_access_denied">You don't have access rights to Books of
Business, used for synchronization or the Books are no longer valid. Please check
your Books of Business configuration settings in the Control Panel or contact
technical support for assistance.</str>
<str key="msg_record_access_denied">Access denied.</str>
<str key="msg_record_set_query_save_record_set">Do you want to save changes to
record set?</str>
<str key="conflict_resolution_form-advanced_view">View details</str>
</res_root>
```

d. Replace this code with the following:

```
<str key="msg_book_access_denied">You don't have access rights to Books of
Business, used for synchronization or the Books are no longer valid. Please check
your Books of Business configuration settings in the Control Panel or contact
technical support for assistance.</str>
<str key="msg_record_access_denied">Access denied.</str>
<str key="msg_record_set_query_save_record_set">Do you want to save changes to
record set?</str>
<str key="msg_record_set_book_is_not_selected">The book is not selected.</str>
<str key="conflict_resolution_form-advanced_view">View details</str>

<img key="conflict-
item_resolution_unresolved">iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/
9hAAAABmJLR0QA/wD/AP+gvaeTAAAACXBI
WXMAAAsTAAALEwEAmpwYAAAAB3RJTUUH1wweDjIRO4cSEwAAAQVJREFUeNrdkbFKA0EURa/
iZ2wl5Adis/MFNuKGCBb2NqJhsbBYZIpFxMpGIUVqFYzoZ4w/
YeV8whJ1Zt+7KUxgXWRtbPSVw7uH8+4Af31Wfh/JDbDug9ba1FqbLN99D3xZB/
M8T5v7q22AEKgVOHkqnaqaoigS3wM/BAgCqKrpBJRb1tQCRAWOHs7uD6bnr8vwOBs5kkknAIC/
yE53gwBRgFbYk3zuNihLr6quGV6oQ1UdSfeTAZraQYD3Gti5uzYhBFRVhU5As7BxNnKTwaGbRWAWgc2
byWWMsbvEb26e3g73j9/qTxMR+fKNa23A1XZuVNWQTBaFuRACHgd7XkRSkh7/
a+bJIrXmJ8rf7AAAAABJRU5ErkJggg==</img>
<img key="conflict-
item_resolution_left">iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9hAAAABmJLR0QA/
wD/AP+gvaeTAAAACXBIWXMAAAsTAAALEwEAmpwYAAAAB3RJTUUH1wweDwUKICcR/
AAAAQhJREFUeNrtk7tKxFAQhv+cI6y9hYW7D2BhGTD12u22K9jlNfIOYhNIZbZLtb6F4MHGaylEZE+v
eNt1z2Z+ixDRxIXU4jRzYebjH4YB/s1r2zi+++DO000PAHzft1VdtQWcnt/
j6Hp9mud5kCRJd6WCg+NbFoWAIhAhpPKFfMV9dbbvnDNRFNkGYHBo+GOgELCWiwj6nRKyVge8PL7Wmk
sV/FYbbFyY5ZJdAGgA9jomIBmQDErJApI4edgeiQiGm5fT+dyzWmurtW4CFouFJWlI2gogInh/
fhsNt64msxmhlJporU0cx7b1GcMw3PU8rwcASimTpqn9dYVV5pyzACwAZFlm/9AvfAJw471I3N8/
hQAAAABJRU5ErkJggg==</img>
 <img key="confirm_page-accept">iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/
9hAAAABGdBTUEAALGOfPtRkwAAACBjSFJNAAB6JQAAgIMAAPn/
AACA6AAAdTAAAOpgAAA6lwAAF2+XqZnUAAACTUlEQVR4nGL8//8/
AyUAIICYKNINBAABxEKqBqEKDU0+cd45QkJiWufitwoCBBDRBgjmG4gysv1ZI6MqrKRrICvz7M7fdyB
xgAACe0GqyFxRMt9cHJdmgWxjMx5BngNOTrp2yV7WMl+eMP27ePF6IkgOIIDALjDQ0bv67tOHm+JpFj
4vZ514CtMomWLF+p3xV6qhmUaPq604p6wgO8PWk88Zjh2/
mfmu+8omkBqAAAK7QJxT4LObg6YBvzDvSplkOyGQmESircY/NsZlTm6aU5M9dTn1xIQZdp16w7D/
4J36191nZ8EsAQggsAF7D56ayfGXhSEz2tWaiYNpnVSMgyufGOeumCiLkFQHKwYBLjaG9afvM+w+fLv
+Ze+RJmTvAQQQIygdSMc7qKuoi1+ozvLn2HTq+P+7N79+d3KS4dIXV2T48ucTw+4rNxm2rbq74+GsnZ
7o4QMQQOAwYP/LdufejTeLD1y6nBpm6sB4W/+ac4eLD1b7C/8b4/ac4eD5wybF9958OXp+Cv
+wN0wWNIqonmTV4lXTI8Ye/ziSY2+Zu/ykaPGfC/q4eXHQiGkntwhXUWd0OXDUq1HqT2T48ucTw+4rNxm2rbq74+GsnZ
7o4QMQQOAwYP/LdufejTeLD1y6nBpm6sB4W+khlwyPCMObH68Yzt8D+nnzq4sMH/
+kYgtggAACe+Hekl1/GT4zrzt3/NH/p59fMfAyczI8+/ac4eLD5wybF99580Xp55CHK/
c+wWYAQADBUyLnH479j6993Xfj+ROGN3/eMFx9/ozh6Lo3b/
6++hd2d83O07iiGCCA4Abc2rzp579Hf7deOv6eG7twxnN3z+8v7C+5jbGzefxqUZBAACCCUlcjCyL
3p+4Vf+xydvJV+f+JF99cimnfg0gwBAADGi50Zzt+i2f78YP54+sKSTkGYQAAggDANIBQABBgB3CeGk
8W9wnQAAAABJRU5ErkJggg==</img>
 <str key="application_tb_caption">$app_name$</str>
 <img
key="animated_daisy_16x16x12">iVBORw0KGgoAAAANSUhEUgAAAMAAAAAQCAYAAABA4nAoAAAAB
GdBTUEAALGOfPtRkwAAACBjSFJNAAB6JQAAgIMAAPn/
AACA6AAAdTAAAOpgAAA6lwAAF2+XqZnUAAAACXBIWXMAAA7DAAAOwwHHb6hkAAAAGXRFWHRTb2Z0d2F
yZQBQYWludC5ORVQgdjMuNS44NzuAXQAAD6dJREFUaEPt2teLLNUWBnD/
ANH74rOlCEbwqChiHEVUBLUNqBjHiAH1qGBEGSMmbDPmNos5Y7YNmHPOjlkxldnHuutXzBq2bZfeO17
vkwc23Weqvupdu75vrW+tXQss8Af/Vlttb7xR+f80bHNN9+8b8wVv8suu/SNueIPOCAvjFX/
NFHH9035oo/44wz+sZc8Zdddlnfmcv+lltu6RtzxT/00EN9Y674F198sW/MFT89Pd035oqv67pv/
Mf4K6+8stpoo43q7bbbbj7QYost1lRV1fj+3Xffzfvmm2+GP/
zww7yuC95www3VtttuW++9994tfo011mhWX331Fr/nnntOLLLIIsOpqamJLvwdd9xRBbY+9NBDW/
zGG2/uGL4ffvhEyuuuOKw9OpqamJLvwdd9xRBbY+9NBDW/
zGG2/cGL4ffvjhEyuuuOKw9OpqamJLvwdd9xRBbY+9NBDW/
fiX/ssceqU045pT733HNbfIInmT9/fos/88wzJ7bZZpv6wfI4MmqTBrZY+tpCTbwJLv19IwDb6X7n/
8GBFTbNB6UzUnkc9V3vVTzKVHz9Fd5dpxS3yNvecBh7+oSMWw4iY91zpL+XpEDdyv+/
9Fgo4B/7/GFq8vrTKYMbWhVYYoFDZkiQw3bhFsyU1v0iq0gojbhuI6aY2c3nO7xPR55UmHDB+SEhnmBy5Y8
```

```
d16qmnNkT01ltvVaUADj744Ontt9++ueqqqzrx5jwYDJoQS/3BBx9UZ599dmNY6xDwtLlERO/
EE9Ctt97axKg/++yzqhRAkGP6+OOPb+6///5OfJCrinVvYtRff/11FaJpDL9/
7bXXTse9N0888UQnPvhRvfnmm02MOgJlVQrg3nvvnb7xxhubEFcn/
ueff67id5sYdfDsX6UAYm2mY32b9957rxMfwbVaaKGFmgUXXLCOdapKAQRuOtan+eKLL36PFxWTGFtu
ueUA0Xq93kAGWHzxxRuKRH4ZwGK4UWpPEYiKSYwg3wDRJicnBzLAmmuu2Wy11VZTyL/88su/
AONGqT3xomIS46CDDhog2v777z8ggE022aSJrDBlThtuuGGLd6PUnnhRMYkRAhkgWpwzkAGMIPWUOW2
99dYtntBlu8SLikmMiPQDRItzBjJACLYJ4kyZU2SxFk/
osl3iRcUkxuWXXz5AtEsvvXRAADGf5qyzzpoyp8MOO6zFE7psl3hRMYkR5B0g2s033zzIDBDXmjKnk0
46qcUTumyXeFExifHwww8PEC0C0IAArrnmmibOnzKnCy+8sMUTugCTeFExifHSSy8NEO2FF14YZAa47
777pszp+uuvb/
GELtslnivIwBjCHSBaWJdBCuCpp56aMqdY4xZP6MSaeK4gA2NwbYBrwa9BCuCNN96YMqf4zRYPK9sln
ivIwBh8G8Tfm1VXXXWQAvjkk0+mzCnWqMUTumyX+AVYgyRIPNjJ5ZZbrhaFww71jbjQkDJ//
PHHnhukcqkuL8AaJEHiwU6uvfbadUTeOkTQN5ZYYolhqLIJP9tzg64l1SWeNUiCIGLUDfV+++1n9I24
qeEyyyzTBHEm3SCVz5s3bxbPGiRB4sFO7rHHHvVRRx1Vx9/
7Roh5SIzujcBlOVYvf581SILEIk+GYOoICHVE8r6x8847D2M9GvdG4LIcq5d41iAJ8tprr01GFqhF8Q
suuKBvhKiHgWujL4HLcqxe4lmDJAgiRxaog7R1RN2+EQ93GFlJZJ4kcFmO1Us8a5AE+fTTTydvu+22+
sEHH6wjeveNsHTDuA8EmiRwWU6WSjxrkASJIDf5+OOP1yEAWbhvhIiGF198cePeCFyWY/
USLzBmgMQRGSAEUL/
66qt9I4Q0lJncG4HLcqxe4lljpBUgcUQG4Dg++uijvhFCGspM7o3AXYvVSzxrnAESRxZeeOF6lVVWqb
///vu+8e677w5x1r0RuGuxerMC4I/TIojYsUC9uOFZhfzyyy/SSXtjFspn+jwX4Y/
TIgTZRMxeRJ9ZfHjyKorB9sYslM/
0efD8cVqEuE4T1+tFBJzFxwOs9tprr54bs1BSXPo8eP44LUJcp4kF7olyeYMRQVmbHmFbKBYv6xzn8M
dpEU4++WQPpxcLPYsPElbuibAFChYv6xx4/
jgtAqKEfezFQ5vF33777ZV7ImyBQnDJOgeePw5MaxE83CBgLzLCLD6Ewxr1CFugYPGyzoGPSDnxzDPP
tBbBww371ou/zeLjejJWj7AFChYv6xx4/
jhI0loERIkg1wuyzeJDDJV7ImyBgniyzoFXH6ZFFtxwJcg2i2cH3RNhCxSCQ9Y58OrDJZdcsrXIAiWu
4Fw+v48//rhyT4Q9s0azdY5z1IfrrLNOa5EFSlyJoDaLZ+fcE2HPrNFsnZO/
0X6yCqIlwvzmQPzHDVG11MbfjR73f1ZBtESY0eMzIpqW2vi7cXhWQbREmNHjbmjllVeupTb+bhyeVTj
uuOOGCDN6nKDDTtWsnfpmHJ5VEC0RZvQ4Qe+www6y2kB9Mw7PKoiWCDN6nKD33XffmrUr7Ud5HqsQcx
sizCieoGU11o6FGPf7rEIIYYgwo8cJWlZj7dQ34/
CsgmiJMKPHCRpxWbtx83O+oMYtlOTN6xC0rMbaqW/G/
T6rzC0ImKPHCVpWY+3UN+PwrHIEqaGAOXqcoGU11u4396dNmN0SnyzDuIv7m3SWvs6nFpc2YXZL+GWW
oQsvnaWv86nFpU2YxSK/zDJ04aWz9HU+tbi0CXllxaJPlqELz85lZ8unFq82YRaL/
DLL0IVn57Kz5VOLV5uQV1Ys+mQZuvDsXBb2PrV4rWG5pta4C+/ZlM/
Ks7OG5Zpa4y68Z5PPyqdnZw3LNbXGXXjPJp+VT8/
OGuaaqhetcRfesymflWdnDXNNuQ9r3IX3bPJZ+fTsrGGuqXrRGnfhPZvyWbUt3v+lACzqXxGARf27Be
AhZXdrVAAW9c8E4CFld2suAvCQ8oH9FQHobs1FAIJUdrfmIgBBKkUwKgBC+CsCIIQ/E4AglSIYFYB1/
a8FkGqRVqVXabb0z3lc2lPMSNPjFCatSq/SbOmf81xpWXqWpsfhXV96lWZL/5znSlvSszQ9Di+tSq/
SbOmf89wZCzYsa4/yOtKq9CrNlv45z5GWpeeZjtbvpmB+0qs0W/
rnPFFalp6l6XHzZyvZSzaz9M95LlvKnrKp4/
BsJXvJi4+zIGwpe8qmjsOzley14rH0z3kuW8qesqnj8Gwle8lmlvVjnsuWsqds6jg8W8lespll/
ZjnsqXsKZs6Ds9WspdsZlk/
5rlsKXvKpv4Or6BSWCmwFFoKrtJHI5QiRoGmCNHTVbgVD3e+wkpkUWjFJBUcVR5HKEWMAk0RoqercMv
jCiqFlQJLoaXgKn00QiliFGgz3YphhVviFVSEo8BSaJlr6VMRSmGmQFOEI4nCLfE2jvTO3ZtCy1xLH4
1QCjMFWmAaBZvCLfE2jvTO3ZtCy1xLn4lQCjMFmkJNwaZwS7yGQtlgMNeSxAiliBchFeEaFhoXiSfqs
sFgrqWPRihFvE6WIlzDQuMi8RoKGgvuTaPBXEsfjVCKeJ0sHNGw0LhIvIaCxoJ702gw17KOFFBxSoMC
RzQsNC4Sr6GgseDeNBrMtawjBVRFvAYFjmhYaFwkXkNBY8G9aTSYa1lHCqg4pUGBIxoWGhezQtBS01r
TYtNq03Kj1mzDiaqIpY2ll4soWnd5AeTVWtNi02rTcqPWbMOJqoiljaWXiyhad4nXUtNa02LTajNRas
02nKiKWNpYerlEoHWXeC01rTUttplWGz8+zDacqIpgWnRlyy7xWmrErcVmjoSq9ZZtOIJBMC26mb2MZ
qZ1115CSw1Gi80cCVW2yjacqIpYWnT2MhBF6y5/
X0t5pvsyMMeZTsow29CiKmJp49rLQBSt68RrKcNoMZsjocpW2YYWVRFLG9deBqJoXSfe7qnWshazORK
qbJVtaFEVsbRx7WUQgdZ14rWUtZa1mM2RUGWrbENzFQIrbtnLIAKt68RrKWstazGbI6HKVtmG5ioEVt
yyl0EEWteFgGqtZS1mcyRU2Srb0IKjwIpb9jKIQOt6VgA2VWyu+ENOUKrKjRjWAgjBnGPTptzSln7SN
uUEpar0a6wF0SAYPJKWW9o2VWyuOJYTlKpyI4a1IBoEc45Nm3JL28ZIdkZSoKxaFpesBdHYw4C3aVO+
0mFTJTdmUqCsWhaXrIXMgWDwNm3KVzpsqthcmbm3VqCsWhaXrEUcawnmHJs25SsdNhVtLjqWAhXV1St
8NWtBNAjmHJuW5SsdNhVtLjqWAmXV1CszvrrdnEQw59i0LF/
psKloc9GxFCirlvUKa0E0COYcm5blKx02FW0uOpYCZdVyI5K1JhoB1jk2LctXOmwq2lx0LAXKqqlX1J
WsNdEIsM5B8vKVDpuKNhcdS4GyalmvsNZEI8A6x6bl2Fc6TCxTlBNzK56/piJRtqsFlxPLFOX/
Wazw1yyGKFvuADqn/GdimaL8Pbfi2QkWQ5RNoo1i/Z8w06L5fwqAv5YBRNkk2jg8YaZFczwFwF/
LAKJsEm0cnjDTojmeAuCvWQxRNok2Dk+YadEcz2Kdv2YxRNkk2jj8zG5qa9Ecz2Kdv2YxRNkk2jg8Ya
ZFczyLdQGOxRBlk2jj8ISZFs3xfBUFZ1hsLiMD7Tg8YaZFczyLdfUli81ljKtP81qEmRbN37JYZ4e5F
C4jA+2437dF3r5wlf6sfBmLz2Y1/
ojAVCk1pT8r30Xhs1mNcgt7dBJUKTWlPytfxuKzWY3fbGGPXEBWYs2yPinfReGzWY3yFY7R35eVWLOs
T8p3UURk4PIVjlG8rMSaZX1SvovCZ7Ma5Ssco3hZSQbI+kT0N5zHZ7Ma5Ssco3hZiTXL+kT0N5zHZ7M
a5Ssco3hZiTXL+qR8F4vPtntdvsIxipeVWLOsTzIDOE+dyWqXr3CM4mUl1izrE9HfcJ6NOyTu2kNxjq
zEmmV9Ur6Lpc5ktctXOMaKoPzjP6/j/
```

```
p9fxx15Iv+8jv73v47+b2iJLDfdg8z0AAAAAElFTkSuQmCC</img>
<str key="page-sync-show_control_panel_button">Show Control Panel...</str>
 <img key="general_conflict_page-page_image"
>iVBORw0KGgoAAAANSUhEUgAAACAAAAAgCAYAAABzenr0AAAAGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbW
FnZVJlYWR5ccllPAAAA9ZJREFUeNrsV01oU1kUPnkmqfalMVozqWlKMRa16kLciAyhtDrTjFIZ0Sq0W
jBgoahFdCOCggtBGSoirgQ7zIzjoII4lEhqnayULLTt09RAtDZR21Rbokmrr03eTz3nmagD7WuisS70
wiHJufee77vnfPcnmomJCfiajYGv3L4T0HyJRR069Odaq9V+SRSTPQcOVP2KvvGZygCzb99Zh9lc0l6
5c3WxRqOtRl/hTJWAaWo67SguXuL+ybWGZU1zQBAE8htnggDT2NiigG9o/
JE1IDjg7hZFkfq0X1oDjMt10lFSssy9ee961liYDxJWfEIG+KfFrToxkeAvfy4BpqHhuMNmW+qubXay
c39gIRlHcPHDgLwFU09uPXJNPT3YZp865b2m1epXRiJ920+caPChT06D19cfc1gsdveWPdWs0czC2Av
sTP4/
gPh06uCkkek0UEhKJkWTsknhH+mmwGxe5K7ZVcWaigzAI5AQA5D4zC0TAkYaRIomZZPISOkpEoWiKLE
P74eh71YUEi9xtXx2RrFnTUNgYWVlfdNKx2LQz9ZB6YqF+sHg69qystW3OjtvjOl0cywaYe6KBJ8EvV
4POp1OUX6mxnEcaA8fvnwpLy9/m1qd0s0wLx9qmirYq2cEd03N3h1tbWf/
TiTGCgRho5MC2u12MJlM78cHAgHV1VFsLcPotu08ujFj2RfMz4fNzetYoUW4MD7+xtXR8ft5WZZBkjY
4KWBpaSkYje/Onu7ubtVYQ0MhTkuTpPEsLg+svnE+C7X7q1nhN6FVkmSX1/vHeUmSMK2/
KJmw2WxgMBiUFV68ePAMTnszSShCDSgEkrEsN78OSSwwwKbdVSzOb0UCa+7du35OlomEUyFhsVjS5fO
i9U0Shk6LmEJAiGV/Ag32RiH0JAy0E1LB7vj97VgKGX0/OxmGSRMgcP9UcRQCYzEhK/
BoNArBYBBCoRDWuf1K6nB6TiQCgQ6gcoyMxNbxfDyCPkktlkLA5/OpApaXl7//Ho/
HFeBwOAydnW7P3bv//oXuV6ksKCSCQa+A1p76PaJKIBJ5xGHNVqkNslqtyufo6CgMDAxAf38/
dHVd93R1tZ0jsVMtP6orgdKKaA8mKGHT3YZb0ZbTuT9JP1tX19JcUVEBPM/
D8PCwQoDjPB4SHa02BSh+6m1Gl9FttAdT3Nt2LFFzGpzS7vff8KDYcgKeJjCYskkbCYrAe3t7oafnpg
dFljPwaV8rhB+PDz3jOL7o8WPffyisnIJn8iKyodHtV4T2CI3LJXgmBEiY9KbR0w5MWc7AFQLf/
xt+8wTeCjAAaiUO1zpinAEAAAAASUVORK5CYII=</img>
</res_root>
```

Options for Uninstalling Oracle CRM On Demand Desktop

This topic describes the options that are available for uninstalling Oracle CRM On Demand Desktop. It includes the following information:

- *Uninstalling Oracle CRM On Demand Desktop by Using the Windows Control Panel*
- *Uninstalling Oracle CRM On Demand Desktop by Using the Oracle CRM On Demand Desktop EXE File*
- *Uninstalling Oracle CRM On Demand Desktop from a Command Line*

Uninstalling Oracle CRM On Demand Desktop by Using the Windows Control Panel

You can uninstall Oracle CRM On Demand Desktop by using the Windows control panel.

To avoid a loss of data, it is recommended that all users synchronize and back up all personal data before removing Oracle CRM On Demand Desktop.

To remove Oracle CRM On Demand Desktop using the Windows control panel

1. Make sure that Oracle CRM On Demand Desktop users have synchronized and backed up their personal data as follows:

   a. In Microsoft Outlook, perform a synchronization.

   b. Back up personal data.

      To back up personal data, it is recommended that the user use the standard Microsoft Outlook feature to export personal data to a file.

2. To remove Oracle CRM On Demand Desktop:

    a. Log in to the computer as an administrator.
    b. Click the Start menu, choose Control Panel, and then open Programs and Features.
    c. In the Currently Installed Programs window, click Oracle CRM On Demand Desktop, and then click Uninstall.

> **Note:** You can confirm the removal of all Oracle CRM On Demand Desktop related data by clicking Yes when you receive the following message: Uninstalling the CRM On Demand Desktop application will remove from Microsoft Outlook all your CRM related data, which is not synchronized yet. Click Yes to continue uninstalling the application. Otherwise click No and perform full synchronization. Then make a backup of all shared contacts, which you want to keep in Microsoft Outlook. Try uninstalling the CRM On Demand Desktop application again.

Uninstalling Oracle CRM On Demand Desktop by Using the Oracle CRM On Demand Desktop EXE File

You can uninstall Oracle CRM On Demand Desktop by running the Oracle CRM On Demand Desktop EXE file.

To uninstall using Oracle CRM On Demand Desktop EXE file

1. Navigate to the Oracle CRM On Demand Desktop.5.0.0.x.exe installation package.
2. In the Welcome dialog box, click Next.
3. Select Remove and click Next.
4. In the InstallShield Wizard dialog box, click Finish.

Uninstalling Oracle CRM On Demand Desktop from a Command Line

You can run the Oracle CRM On Demand Desktop.5.0.0.x.exe installation package from a command line interface on the client computer to uninstall Oracle CRM On Demand Desktop.

To uninstall Oracle CRM On Demand Desktop from a command line

1. Open a command line.
2. To unistall Oracle CRM On Demand Desktop, use the following command:

```
"C:\...\Oracle CRM On Demand Desktop.exe" /x /s /v"/L*v \"%temp%\result.log\" /qn"
```

8 Administering Single Sign-On

Administering Single Sign-On

This chapter describes how to administer Oracle CRM On Demand Desktop Single Sign-On. It contains the following topics:

- *Overview of Single Sign-On for Oracle CRM On Demand Desktop*
- *Types of Authentication That You Can Use with Oracle CRM On Demand Desktop SSO*
- *Single Sign-On Services that Oracle CRM On Demand Desktop Supports*
- *Configuring Oracle CRM On Demand Desktop SSO*

Overview of Single Sign-On for Oracle CRM On Demand Desktop

Oracle CRM On Demand Desktop Single Sign-On (Oracle CRM On Demand Desktop SSO) allows you to implement single sign-on for the Oracle CRM On Demand Desktop client. It supports standard Web browser functionality, such as HTTP forms, cookies, and process redirects. It provides the following capabilities:

- **Supports your existing Web single sign-on feature.** You can customize Oracle CRM On Demand Desktop SSO so that it works in your network topology and for mobile or remote access.
- **Supports your custom user interface.** You can use Oracle CRM On Demand Desktop SSO, which is automated and transparent to the user. You can also use it with a custom-authentication screen that you already use, or you can include it with your company branding.
- **Supports typical, login name and password authentication or custom authentication, which requires more than a login name and password (multifactor authentication).**
- **Handles single sign-on, session, and network errors in a way that is transparent to the user.**

This topic describes an overview of single sign-on for Oracle CRM On Demand Desktop. It includes the following information:

- *Types of Authentication That You Can Use with Oracle CRM On Demand Desktop SSO*
- *Single Sign-On Services that Oracle CRM On Demand Desktop Supports*

Types of Authentication That You Can Use with Oracle CRM On Demand Desktop SSO

This topic describes the types of authentication that you can use with Oracle CRM On Demand Desktop SSO.

Noninteractive Authentication

Noninteractive authentication is a type of authentication in Oracle CRM On Demand Desktop SSO that reuses the login name and password from the Oracle CRM On Demand Desktop Login dialog box. Oracle CRM On Demand Desktop SSO does not display a separate CRM Desktop SSO Login dialog box in the client. Noninteractive authentication includes the following functionality:

- **Save password.** The user can set the Save Password option on the Oracle CRM On Demand Desktop Login dialog box to one of the following values:
    - **Include a check mark.** Oracle CRM On Demand Desktop SSO stores credentials in an encrypted form in the Windows Registry. It does not prompt the user for credentials the next time that the user starts Oracle CRM On Demand Desktop.
    - **Do not include a check mark.** Oracle CRM On Demand Desktop SSO stores credentials in memory but not in the Windows Registry. It prompts the user for credentials the next time the user starts Oracle CRM On Demand Desktop. It prompts the user for credentials once for each SSO session. This behavior typically occurs during the first synchronization that occurs after the user restarts Oracle CRM On Demand Desktop.
- **Detect session expiration.** If an Oracle CRM On Demand Desktop SSO session expires, then Oracle CRM On Demand Desktop SSO reestablishes the session without involving the user. Oracle CRM On Demand Desktop SSO requires user interaction only if the user credentials are not valid.
- **Use only the login name and password.** Noninteractive authentication does not support multifactor authentication, such as requiring code from a security token in addition to the password.
- **Detect invalid user password.** The SSO script can detect if the user enters an invalid password and then react accordingly.

Interactive Authentication

Interactive authentication is a type of authentication in Oracle CRM On Demand Desktop SSO that displays a separate Oracle CRM On Demand Desktop SSO Login dialog box in the client. The user must enter the login credentials in this dialog box. Interactive authentication includes the following functionality:

- **Ignore save password.** The user can set the Save Password option on the Oracle CRM On Demand Desktop Login dialog box, but Oracle CRM On Demand Desktop SSO ignores this setting.
- **Detect session expiration.** If an Oracle CRM On Demand Desktop SSO session expires, then Oracle CRM On Demand Desktop SSO prompts the user to provide credentials and then reestablishes the session.
- **Multifactor authentication.** Supports more than the login name and password for authentication, such as requiring a code from a security token in addition to the password. The Oracle CRM On Demand Desktop SSO Login dialog box can include more fields, images, a list of questions which, the user must answer, ActiveX controls, and other items that your implementation requires for authentication. It can support an input field for an RSA (Rivest Shamir Adleman) token, a CAPTCHA, or other information, which only the user can provide.
- **Supports Internet Explorer.** The Oracle CRM On Demand Desktop SSO Login dialog box is an Internet Explorer ActiveX dialog box.

Interactive authentication requires Windows Internet Explorer 7 or later to be installed on the target computer.

Single Sign-On Services that Oracle CRM On Demand Desktop Supports

Oracle CRM On Demand Desktop supports SAML Web SSO (Version 1.1 and Version 2.0) with either POST or Artifact profiles. Authentication must be browser-based (for example, form-based, HTTP Basic, HTML, and so on).

Configuring Oracle CRM On Demand Desktop SSO

Oracle CRM On Demand Desktop is packaged with an SSO installation, which is embedded in the Oracle CRM On Demand Desktop EXE file. It includes the SSO scripts for the Generic SAML Connector.

By default, SSO is disabled and must be enabled. This procedure shows you how to enable SSO in the default configuration of Oracle CRM On Demand Desktop by setting the COMPANYSSOID parameter, using the Oracle CRM On Demand Desktop installer on the command line.

Note: If Oracle CRM On Demand Desktop is already installed, you can enable SSO by setting the CompanySSOID registry key in the [HKEY_CURRENT_USER\Software\Oracle\CRM OnDemand Desktop \Connector directory in the Microsoft Windows registry.

To configure Oracle CRM On Demand Desktop SSO

1. On the command line, set the COMPANYSSOID parameter, using the Oracle CRM On Demand Desktop installer:

   ```
   "Oracle CRM On Demand Desktop.5.0.0.x.exe" /V"COMPANYSSOID=<SSOID>"
   ```

   In the command, <SSOID> is the company sign-in user ID from the Oracle CRM On Demand Company Profile page.
2. Select Use Single Sign-On from the Oracle CRM On Demand Desktop page to enable SSO.

9 Administering Oracle CRM On Demand Desktop

Administering Oracle CRM On Demand Desktop

This chapter describes how to administer Oracle CRM On Demand Desktop before your end users install Oracle CRM On Demand Desktop for use with Microsoft Outlook. It contains the following topics:

Before Administering Oracle CRM On Demand Desktop

After Oracle CRM On Demand Desktop licenses are assigned to your company, the Manage CRM Desktop privilege is visible within the Administrator role. You must also provide access to Oracle CRM On Demand Desktop for your users, see *Allowing Users to Access Oracle CRM On Demand Desktop within Your Company*.

It is recommended that you create new roles for specific Oracle CRM On Demand Desktop users. This action might help control license consumption. For example, if you have 400 sales representatives, and you must assign Oracle CRM On Demand Desktop licenses to 200 of these representatives, then it is recommended that you create a new role for the Oracle CRM On Demand Desktop user. You can then add these 200 representatives to this new role, and assign the Oracle CRM On Demand Desktop licenses to only those 200 users.

Allowing Users to Access Oracle CRM On Demand Desktop within Your Company

You must provide access to Oracle CRM On Demand Desktop for your users.

To allow users to access Oracle CRM On Demand Desktop within your company

- Assign the Enable CRM Desktop privilege to each Oracle CRM On Demand Desktop user's role.

  For more information on privileges, roles, and users, see *Oracle CRM On Demand Online Help* .

Changing the Behavior of Oracle CRM On Demand Desktop

This topic describes how you can change the behavior of Oracle CRM On Demand Desktop. It includes the following information:

- *Modifying the Windows Registry to Change Oracle CRM On Demand Desktop Behavior*
- *Setting Behavioral Limits for Oracle CRM On Demand Desktop*
- *Controlling the Object Types That Oracle CRM On Demand Desktop Displays inthe Filter Records Tab*
- *Controlling the Fields That Are Available in a Filter*
- *Controlling How Oracle CRM On Demand Desktop Converts Contacts*
- *Controlling How Oracle CRM On Demand Desktop Deletes Records*
- *Controlling the Exclusions List*
- *Controlling the Confirm Synchronization Tab*
- *Controlling How Oracle CRM On Demand Desktop Handles an Archived Item*
- *Modifying the Windows Registry Parameters That Affect Local Synchronization*

In addition to these topics on changing the behavior of Oracle CRM On Demand Desktop, you can also further customize how fields and other objects appear and behave in the user interface, and set up book support for object types. For more information on this customization, see *Oracle CRM On Demand Desktop Customization Guide* .

Modifying the Windows Registry to Change Oracle CRM On Demand Desktop Behavior

You can modify several Windows Registry parameters that affect Oracle CRM On Demand Desktop behavior. For example, you can specify the visibility of certain dialog boxes, the path where certain files are stored, or if the live update feature is allowed.

> **CAUTION:** Modifying the Windows Registry can cause serious and permanent problems that you might not be able to resolve. Make only the modifications that you require, and make sure that the modifications do not negatively affect system functionality or performance.

For more information, see *Changes That Oracle CRM On Demand Desktop Makes During an Installation*.

To modify Windows Registry parameters to change the behavior of Oracle CRM On Demand Desktop

1. To automate changes to Windows Registry parameters, use an administrative tool, such as Systems Management Server or BMC Configuration Management.
2. Add or modify Windows Registry parameters, as necessary.

   For more information, see *Changes That Oracle CRM On Demand Desktop Makes in the Windows Registry*.

Setting Behavioral Limits for Oracle CRM On Demand Desktop

You can set the limits that affect certain Oracle CRM On Demand Desktop behavior. The files that this topic describes are part of the customization package. To modify one of these files, you can use any editor that supports editing in JavaScript or XML, such as Notepad.

To set the behavioral limits for Oracle CRM On Demand Desktop

1. Modify the helpers.js file to set certain limits.
   The helpers.js file is located in the customization package delivered in the installation ZIP file. Use the values from the following table.

Variable with Default Value	Description
var max_attach_file_size = 20	Sets the maximum size in megabytes of an attachment file in Microsoft Outlook.
var ask_delete_confirmation = true	Determines if Oracle CRM On Demand Desktop displays the Revert Deletions and Accept Deletions buttons in the Confirm Synchronization tab of the Synchronization Control Panel.
var action_selection_limit = 30	Sets the maximum number of items that Oracle CRM On Demand Desktop can process when the user executes a toolbar command. For example, if you set var action_selection_limit to 30, then the user can choose only 30 records in Microsoft Outlook, and then execute a command by using the toolbar in Microsoft Outlook. An example command is Email to Contacts. To avoid poor performance, you can modify this limit.

2. Edit the od_meta_info.xml file, using values from the following table. For more information on customizing, see *Customizing Oracle CRM On Demand Desktop*.

Variable	Description
max_commands_per_batch	Sets the maximum number of commands for each batch. For more information, see *SiebelMetaInfo Element of the od_meta_info.xml File*.
max_ids_per_command	Defines the maximum number of object IDs. For more information, see *SiebelMetaInfo Element of the od_meta_info.xml File*.
open_with_url_tmpl	Sets a template for the code that Oracle CRM On Demand Desktop uses to create a URL to open the page in Oracle CRM On Demand for that record.
ViewMode	Sets the visibility of an object. Sales Rep is the only view mode supported in data synchronization. For example: `ViewMode = Sales Rep` For additional information on using the ViewMode attribute, see *About the View Mode Attribute When Using Books*.

About the View Mode Attribute When Using Books

Book parameters are passed using the <Command> element attributes. For example, the following SOAP client request, from Oracle CRM On Demand Desktop to Oracle CRM On Demand, passes the following attributes:

```
ViewMode="Context" BookId="1QA2-1WPK6M" IncludeSubBooks="false"
```

When a user selects a book for a record type, the ID for that book and the IncludeSubBooks value are stored in the Microsoft Windows Registry. The connector then reads these values and inserts them into SOAP requests generated for that particular type. Also, Context viewmode is always used for such requests when you have to provide book parameters in those requests.

For example, the following is a sample SOAP request sent by the client:

```
<soapenv:Header>
 <ClientName xmlns="urn:crmondemand/ws">Oracle, CRM On Demand Desktop</ClientName>
 <crmod:SessionKeepAlive>true</crmod:SessionKeepAlive>
</soapenv:Header>
<soapenv:Body>
 <ClientSyncService_Execute_Input xmlns="urn:crmondemand/ws/crmdesktop/
clientsyncservice/">
 <ListOfCommands xmlns="urn:/crmondemand/xml/clientsyncexecuteinput">
 <Batch>
 <Command ViewMode="Context" BookId="1QA2-1WPK6M" IncludeSubBooks="false">
 <Id>{27FA29C3-AD92-485E-BAE5-5E4567B03020}</Id>
 <Message><SiebelMessage IntObjectName='Account'>
 <ListOfAccount startrownum='0' pagesize='100'>
 <Account operation='querypage'>
 ...
 </Account>
 </ListOfAccount>
 </SiebelMessage></Message>
 </Command>
 </Batch>
 </ListOfCommands>
 </ClientSyncService_Execute_Input>
</soapenv:Body>
</soapenv:Envelope>
```

Controlling the Object Types That Oracle CRM On Demand Desktop Displays inthe Filter Records Tab

By default, Oracle CRM On Demand Desktop displays the following objects in the Filter Records tab of the Synchronization Control Panel dialog box:

- Contacts
- Accounts
- Opportunities
- Activities
- Leads

You can customize the Synchronization Control Panel dialog box to display or not display these object types.

To control the object types that Oracle CRM On Demand Desktop displays in the Filter Records tab

1. Use an XML editor to open the connector_configuration.xml file.

   For more information, see *Files in the Customization Package*.
2. In the connector_configuration.xml file, locate the type element for the object that you must display or hide, for example:

   ```
   type id="Opportunity"
   ```
3. Find the view element for the type you located in Step 2.
4. Set the suppress_sync_ui attribute to True to hide the object or False to display the object.

   For more information, see *View Element of the connector_configuration.xml File*.
5. Save your changes and then republish the customization package.

   For more information, see *Creating and Publishing the Customization Package*.
6. (Optional) To add a new object to the list of objects that Oracle CRM On Demand Desktop displays on the Filter Records tab, do the following:

   a. In the XML code, add new type and view elements for the new object.
   b. Set the suppress_sync_ui attribute for this new object to False.

   Service requests are an example of a new object.

Controlling the Fields That Are Available in a Filter

To control the fields that are available in a filter, you use the IsFilterable property of the field in the od_meta_info.xml file.

To control the fields that are available in a filter

1. Use an XML editor to open the od_meta_info.xml file.

   For more information, see *Files in the Customization Package*.
2. Locate the first instance of the element that defines the field you must modify.

   For example, in the Contact.Account object, the following element defines the Account Status field:

   ```
   <field Name='Account Status' Label='Account Status' DataType='DTYPE_TEXT'
   HasPicklist='yes' PicklistIsStatic='yes' PicklistCollectionType='ACCOUNT_STATUS'
   PicklistTypeId='PickList_Generic' IOElemName='AccountStatus' />
   ```
3. Do one of the following:

   - To make the field not available in a filter criterion, add the following property to this element:

     ```
     IsFilterable='yes'
     ```
   - To make the field available in a filter criterion, add the following property to this element:

     ```
     IsFilterable='no'
     ```

Note that the DataType property must not be DTYPE_ID.

4. Repeat Step 2 through Step 3 for each of the other objects you must modify.

   For example, the Contact.Account object also includes the account status.

Controlling How Oracle CRM On Demand Desktop Converts Contacts

By default, the CRM On Demand Desktop Assistant prompts the user to convert Microsoft Outlook contacts to Oracle CRM On Demand contacts. You can suppress this feature by completing the following procedure. Suppressing the feature leaves the contact records unconverted.

To control how Oracle CRM On Demand Desktop converts contacts

1. Use an XML editor to open the platform_configuration.xml file.

   For more information, see *Files in the Customization Package*.

2. Add the following code to the platform element:

```
<initialization_script>
 <![CDATA[
 application.settings.set("FRA:SuppressConvertItemsMsgBox", "true");
 ]]>
</initialization_script>
```

Controlling How Oracle CRM On Demand Desktop Deletes Records

You can control how Oracle CRM On Demand Desktop deletes records during a synchronization. The delete confirmation feature allows the user to cancel, during synchronization, a deletion that the user made in Microsoft Outlook. If you enable this feature, then Oracle CRM On Demand Desktop does the following work:

- Displays the Confirm Synchronization tab on the Synchronization Control Panel dialog box.
- Uses the Confirm Synchronization tab to allow the user to confirm the delete operation. If the user deletes records in Microsoft Outlook, then Oracle CRM On Demand Desktop displays the Confirm Synchronization tab during synchronization. If the user confirms, then Oracle CRM On Demand Desktop removes the deleted records from the Oracle CRM On Demand database on the Oracle CRM On Demand server.

To control the Confirm Synchronization tab

1. Use an XML editor to open the connector_configuration.xml file.
2. To display the Confirm Synchronization tab on the Synchronization Control Panel dialog box, you do the following:

   a. Add the following code to the root element of the connector_configuration.xml file:

   ```
   <features deletion_confirmation_mode="enable"/>
   ```

   For more information, see *Setting the Delete Confirmation Mode Attribute*.

b. Specify the objects that Oracle CRM On Demand Desktop displays in the Delete from On Demand in the Confirm Synchronization tab.

   For more information, see *Specifying the Type of Object the User Can Confirm for Deletion*.

3. To suppress the display of the Confirm Synchronization tab on the Synchronization Control Panel dialog box, add the following code to the root element of the connector_configuration.xml file:

```
<features deletion_confirmation_mode="suppress"/>
```

How the Number of Deleted Records Determines Delete Confirmation

The number of records that the user has deleted determines whether or not Oracle CRM On Demand Desktop displays the Confirm Synchronization tab. For example:

- If the user deletes three or more accounts, ten or more contacts, or five or more opportunities, then Oracle CRM On Demand Desktop displays the Confirm Synchronization tab.
- If the user deletes only one or two accounts, then Oracle CRM On Demand Desktop does not display the Confirm Synchronization tab.

The following table lists the minimum number of records that the user must delete to cause Oracle CRM On Demand Desktop to display the Confirm Synchronization tab.

Object	Number of Deleted Records
Account	3
Account.Account_Note	10
Account.Contact.Association	20
Action.Contact.Association	100
Action.Employee.Association	100
Activity	20
Attachment	10
Contact	10
Contact.Contact_Note	10
Currency	Not applicable
Defaults	Not applicable
Employee.Position.Association	Not applicable
Internal_Product	Not applicable

Object	Number of Deleted Records
Opportunity	5
Opportunity.Assignment_Group.Association	5
Opportunity.Contact.Association	20
Opportunity.Opportunity_Note	10
Opportunity.Opportunity_Product	5
Sales_Method.Sales_Cycle_Def	Not applicable
User	Not applicable

Setting the Delete Confirmation Mode Attribute

To control the Confirm Synchronization Tab on the Synchronization Control Panel dialog box, you use the deletion_confirmation_mode attribute of the connector_configuration.xml file.

The following table describes the values you can use for the deletion_confirmation_mode attribute.

Value	Description
suppress	Disables delete confirmation. Oracle CRM On Demand Desktop does not display the Confirm Synchronization tab in the Synchronization Control Panel.
enable	Enables delete confirmation. Oracle CRM On Demand Desktop displays the Confirm Synchronization tab in the Synchronization Control Panel. It displays the Revert Deletions button and the Accept Deletions button.
revert_only	Oracle CRM On Demand Desktop displays the Confirm Synchronization tab in the Synchronization Control Panel, but enables only the Revert Deletions button. It displays but does not enable the Accept Deletions button. This is the default setting.
user_confirm	Oracle CRM On Demand Desktop displays the Confirm Synchronization tab in the Synchronization Control Panel, but displays only the Accept Deletions button. It displays but does not enable the Revert Deletions button.

The following example sets the deletion_confirmation_mode attribute to revert_only. The ellipsis points (. . .) indicate code that is omitted:

```
<root>

<features deletion_confirmation_mode="revert_only"

. . .

</features>
```

Specifying the Type of Object the User Can Confirm for Deletion

You can specify the type of object that Oracle CRM On Demand Desktop displays in the Delete from On Demand list in the Confirm Synchronization tab. For example, you can specify Oracle CRM On Demand Desktop to display only opportunity records.

To specify the type of object the user can confirm for deletion

1. Use an XML editor to open the connector_configuration.xml file.
2. Locate the object type that Oracle CRM On Demand Desktop must display in the Delete from On Demand list in the Confirm Synchronization tab list.

   For example, for opportunities, locate the following object type:

   ```
   type id="Opportunity"
   ```

3. In the object you located in Step 2 add the synchronizer element or extend the existing one with the attribute-value pair: threshold="5".

   For example, add the following element:

   ```
   <synchronizer name_format=":[:(Name):]" threshold="5">
   ```

   For more information, see the following topic.
4. Repeat Step 2 through Step 3 for each type of object that Oracle CRM On Demand Desktop must display in the Delete from On Demand list.

Example Code for Setting the Synchronizer Element

The synchronizer element in the connector_configuration.xml file controls the type of records that Oracle CRM On Demand Desktop displays in the Delete from On Demand list in the Confirm Synchronization tab list. It includes a threshold attribute.

In the following example, the threshold attribute set to 5 causes Oracle CRM On Demand Desktop to display the Confirm Synchronization tab only if the user deleted five or more opportunities since the last synchronization:

```
<type id="Opportunity" state_field="ObjectState">
 <view label="#obj_opportunity" label_plural="#obj_opportunity_plural"
small_icon="type_image:Opportunity:16" normal_icon="type_image:Opportunity:24"
large_icon="type_image:Opportunity:48"></view>
 <synchronizer name_format=":[:(Name):]" threshold="5">
 <links>
 </links>
 <natural_keys>
 </natural_keys>
 </synchronizer>
</type>
```

The following table describes the values for the threshold attribute of the synchronizer element.

Value	Description
0	Oracle CRM On Demand Desktop does not display delete confirmation for the object type.
1	Oracle CRM On Demand Desktop displays delete confirmation for the object type.
Any value greater than 1	If you specify any value that is greater than one, then Oracle CRM On Demand Desktop does the following:

Value	Description
	• If the value you specify is greater than the number of deleted objects, then Oracle CRM On Demand Desktop does not display delete confirmation for the object. • If the value you specify is less than or equal to the number of deleted objects, then Oracle CRM On Demand Desktop displays delete confirmation for the object.

Controlling the Exclusions List

The Exclusions List allows the user to exclude an individual record from synchronization even if this record matches a defined filtering criteria. Oracle CRM On Demand Desktop does the following work:

1. To identify the records that it must synchronize from the Oracle CRM On Demand server, it uses the following filters:
   - Filters that the user creates
   - Master filters

   Note: Users cannot configure master filters. Master filters are defined in od_meta_info.xml file in the <master_filter_expr> element of object Type.
2. Excludes the records that are in the Exclusions List. It excludes each record in the list only if some other record does not reference this record.
3. If a record is listed in the Exclusions List, and if no other record references it, then Oracle CRM On Demand Desktop removes this record from Microsoft Outlook even if the Remove Local Records option contains a check mark.

To configure the exclusions list

1. Display the Exclusions button on the Filter Records screen of the Synchronization Control Panel:
   a. Use an XML editor to open the connector_configuration.xml file.
   b. Locate the following features element:

   ```
   <features>
   ```

   c. Make sure the following attribute in this features element is set to True:

   ```
   enable_sync_exclusions
   ```

   You can add this attribute to the connector_configuration.xml file if it is not visible. For more information on adding the enable_sync_exclusions attribute, see *Controlling the Confirm Synchronization Tab*.

Turning on Exclusion Functionality

To turn on the exclusion functionality for a specific object, complete the following procedure.

To turn on exclusion functionality for a specific object

1. Use an XML editor to open the connector_configuration.xml file.
2. Locate the object type that you want to exclude.

   For example, for opportunities, locate the following object type:

```
type id="Opportunity"
```

3. In the object you located in Step 2, add the synchronizer element.

   For example, add the following element:

   ```
   <synchronizer name_format=":[:(Name):]" threshold="5">
   ```

   For more information, see *Example Code for Setting the Synchronizer Element*.

4. Repeat Step 2 through Step 3 for each object type.
5. In the actions.js file locate the siebel_item_delete function and the types_options variable inside the following section:

   ```
    var types_options =
    {
    "Mail": {
    "confirmation": false
    },
    "Account": {
    "exclude_allowed": true
    },
    "Contact": {
    "exclude_allowed": true,
    "exclude_supported": function(item) { return item["SiebelShared"]; }
    },
    "Opportunity": {
    "exclude_allowed": true
    },
    "Lead": {
    "exclude_allowed": true
    }
    };
   ```

6. In this section, add the new object type definition, and add set the exclude_allowed property to True.

   When you add the new object type definition, it must be in the following format:

   ```
    "MyObject": {
    "exclude_allowed": true
    }
   ```

Examples of How Oracle CRM On Demand Desktop Uses the Exclusions List

Assume the following:

- Contact 1 references account 1.
- Contact 1 matches a filter, but account 1 does not match a filter.

In this example, Oracle CRM On Demand Desktop synchronizes account 1 because contact 1 references it. It does not add account 1 to the exclusion list.

For another example, assume the following:

- Contact 1 references account 1.
- Contact 1 matches a filter and account 1 matches a filter.
- After the first synchronization, Microsoft Outlook displays contact 1 in the contact list and account 1 in the account list. The contact list is labelled as Contacts, and the accounts list is labelled as Accounts.
- The user deletes account 1 and then Oracle CRM On Demand Desktop displays the following prompt:

Click No to remove the items from filter

- The user clicks No, and then Oracle CRM On Demand Desktop moves account 1 to the Exclusions List. At the next synchronization, Oracle CRM On Demand Desktop synchronizes account 1 because contact 1 references it.

How Oracle CRM On Demand Desktop Automatically Adds Accounts, Contacts, and Opportunities to the Exclusions List

If the user deletes an account, lead, contact, or opportunity in the Explorer view, then Oracle CRM On Demand Desktop displays the following prompt:

Are you sure you would like to delete this item from Oracle CRM On Demand and Microsoft Outlook? Click Yes to delete the items from both applications. Click No to remove the items from filter.

The user can choose one of the following values:

- **Yes.** Oracle CRM On Demand Desktop deletes the record in Microsoft Outlook and then deletes it from the Oracle CRM On Demand database on the Oracle CRM On Demand server during the next synchronization. If you enable delete confirmation, then Oracle CRM On Demand Desktop requests the user to confirm the deletion before it deletes the record from the Oracle CRM On Demand database. For more information, see *Controlling How Oracle CRM On Demand Desktop Deletes Records*.
- **No.** Oracle CRM On Demand Desktop deletes the record from Microsoft Outlook and adds it to the Exclusions List. If this record is associated with another record in Microsoft Outlook through the lookup field, then Microsoft Outlook displays it the next time that the user synchronizes. In this case a lookup field is a field where you can select the relationship between the current record which references another record. For example, in Contact Object, there is a lookup field that refers to the Account Object Type called Primary Account.

  For example, assume the following:

  - A contact references a primary account.
  - The user deletes the account from the Explorer view and then clicks No at the confirmation prompt.

  In this case, Oracle CRM On Demand Desktop removes the account from the Accounts folder, the Accounts field on the Contact form, and from the Accounts MVG dialog box. The user synchronizes, and then Microsoft Outlook displays the record in the Accounts Lookup dialog box and in the Account field on the Contact form.

Controlling the Confirm Synchronization Tab

You can specify how Oracle CRM On Demand Desktop allows the user to handle deletions that occur during a synchronization. The delete confirmation feature allows the user to cancel, during synchronization, a deletion that the user made in Microsoft Outlook. If you enable this feature, then Oracle CRM On Demand Desktop does the following work:

- Displays the Confirm Synchronization tab on the Synchronization Control Panel dialog box.
- Uses the Confirm Synchronization tab to allow the user to confirm the delete operation. If the user deletes a record in Microsoft Outlook, then Oracle CRM On Demand Desktop displays the Confirm Synchronization tab during synchronization. If the user confirms, then Oracle CRM On Demand Desktop removes the deleted record from the Oracle CRM On Demand database on the Oracle CRM On Demand server.

To control the Confirm Synchronization tab

1. Use an XML editor to open the connector_configuration.xml file.

2. To display the Confirm Synchronization tab on the Synchronization Control Panel dialog box, do the following:

   a. Add the following code to the root element of the connector_configuration.xml file:

      ```
      <features deletion_confirmation_mode="enable"/>
      ```

      For more information, see *Setting the Delete Confirmation Mode Attribute*.

   b. Specify the objects that Oracle CRM On Demand Desktop displays in the Delete from On Demand list in the Confirm Synchronization tab.

      For more information, see *Specifying the Type of Object the User Can Confirm for Deletion*.

3. To suppress the display of the Confirm Synchronization tab on the Synchronization Control Panel dialog box, add the following code to the root element of the connector_configuration.xml file:

   ```
   <features deletion_confirmation_mode="suppress"/>
   ```

Controlling the Connection Tab

You can customize the settings for the Connection tab of the Oracle CRM On Demand Desktop-Options dialog box by modifying the customization package. The following procedure explains how to customize what check box is selected by default in the Options dialog box.

To control the Connection tab

1. Use an XML editor to open the platform_configuration.xml file.
2. In the initialization_script-CDATA section, set the following parameters:

   ```
   <initialization_script>
    <![CDATA[
    application.settings.set("ProxyUsage", "2");
    application.settings.set("ProxyServer", "server_address");
    application.settings.set("ProxyServerPort", "8080");
    application.settings.set("ProxyLogin", "login_name");
    ]]>
   </initialization_script>
   ```

   where the ProxyUsage parameter can be:

   - **0.** Direct connection
   - **1.** Use IE settings
   - **2.** Manual proxy configuration

3. Save and close the platform_configuration.xml file.

Setting the Delete Confirmation Mode Attribute

To control the Confirm Synchronization Tab on the Synchronization Control Panel dialog box, you use the deletion_confirmation_mode attribute of the connector_configuration.xml file.

The following table describes the values you can use for the deletion_confirmation_mode attribute.

Value	Description
suppress	Disables delete confirmation. Oracle CRM On Demand Desktop does not display the Confirm Synchronization tab in the Synchronization Control Panel.
enable	Enables delete confirmation. Oracle CRM On Demand Desktop displays the Confirm Synchronization tab in the Synchronization Control Panel. It displays the Revert Deletions button and the Accept Deletions button.
revert_only	Oracle CRM On Demand Desktop displays the Confirm Synchronization tab in the Synchronization Control Panel but enables only the Revert Deletions button. It displays but does not enable the Accept Deletions button. This setting is the default.
user_confirm	Oracle CRM On Demand Desktop displays the Confirm Synchronization tab in the Synchronization Control Panel but displays only the Accept Deletions button. It displays but does not enable the Revert Deletions button.

The following example sets the deletion_confirmation_mode attribute to revert_only. The ellipsis points (. . .) indicate code that is omitted:

```
<root>
 <features deletion_confirmation_mode="revert_only"
 . . .
 </features>
```

Specifying the Type of Object the User Can Confirm for Deletion

You can specify the type of object that Oracle CRM On Demand Desktop displays in the Delete from On Demand list in the Confirm Synchronization tab. For example, you can specify Oracle CRM On Demand Desktop to display only opportunity records.

To specify the type of object the user can confirm for deletion

1. Use an XML editor to open the connector_configuration.xml file.
2. Locate the object type that Oracle CRM On Demand Desktop must display in the Delete from On Demand list in the Confirm Synchronization tab list, for example, for opportunities, locate the following object type:

   ```
   type id="Opportunity"
   ```

3. In the object you located in Step 2, add the synchronizer element.

   For example, add the following element:

   ```
   <synchronizer name_format=":[:(Name):]" threshold="5">
   ```

   For more information, see *Example Code for Setting the Synchronizer Element*.
4. Repeat Step 2 through Step 3 for each type of object that Oracle CRM On Demand Desktop must display in the Delete from On Demand list.

Setting the Synchronizer Element

The synchronizer element in the connector_configuration.xml file controls the type of records that Oracle CRM On Demand Desktop displays in the Delete from On Demand list in the Confirm Synchronization tab list. It includes a threshold attribute.

In the following example, the threshold attribute of 5 causes Oracle CRM On Demand Desktop to display the Confirm Synchronization tab only if the user deleted five or more opportunities since the last synchronization:

```
<type id="Opportunity" state_field="ObjectState">
 <view label="#obj_opportunity" label_plural="#obj_opportunity_plural"
small_icon="type_image:Opportunity:16" normal_icon="type_image:Opportunity:24"
large_icon="type_image:Opportunity:48"></view>
 <synchronizer name_format=":[:(Name):]" threshold="5">
 <links>
 </links>
 <natural_keys>
 </natural_keys>
 </synchronizer>
</type>
```

The following table describes the values for the threshold attribute of the synchronizer element.

Value	Description
0	Oracle CRM On Demand Desktop does not display delete confirmation for the object type.
1	Oracle CRM On Demand Desktop displays delete confirmation for the object type.
Any value greater than 1	If you specify any value that is greater than one, then Oracle CRM On Demand Desktop does the following: • If the value you specify is greater than the number of deleted objects, then Oracle CRM On Demand Desktop does not display delete confirmation for the object. • If the value you specify is less than or equal to the number of deleted objects, then Oracle CRM On Demand Desktop displays delete confirmation for the object.

Controlling How Oracle CRM On Demand Desktop Handles an Archived Item

Oracle CRM On Demand Desktop does not distinguish between items in Microsoft Outlook that Microsoft Outlook archives and items that the user deletes. If Microsoft Outlook archives an item, then Oracle CRM On Demand Desktop interprets this action as a deletion, and it deletes the corresponding Oracle CRM On Demand record in the Oracle CRM On Demand database. To modify this behavior, you use the archive_activity_days variable in the business_logic.js file. If the deleted Microsoft Outlook event contains the following:

- An end date that occurs further in the past than the current system date minus the number of days that the archive_activity_days variable specifies, then Oracle CRM On Demand Desktop does not delete the corresponding Oracle CRM On Demand record.
- An end date that occurs more recently than the current system date minus the number of days that the archive_activity_days variable specifies, then Oracle CRM On Demand Desktop deletes the corresponding Oracle CRM On Demand record.

The archive_activity_days variable is set to 7 by default. For example, assume you leave the archive_activity_days variable at the default value. Oracle CRM On Demand Desktop does the following:

- If a meeting occurred further in the past than the current system date minus 7 days, and if the meeting no longer exists in Microsoft Outlook, then Oracle CRM On Demand Desktop treats the meeting as an archived item. Oracle

CRM On Demand Desktop does not delete the corresponding Oracle CRM On Demand meeting from the Oracle CRM On Demand database.

- If a meeting was scheduled to occur less than 7 days before the current system date, then Oracle CRM On Demand Desktop assumes the user intentionally deleted the meeting, and it deletes the corresponding Oracle CRM On Demand meeting from the Oracle CRM On Demand database.

To control how Oracle CRM On Demand Desktop handles archived items

1. Open the business_logic.js file with a program that can edit JavaScript code.

   For more information, see *Files in the Customization Package*.
2. Change the value for the var archive_activity_days attribute.
3. Add your modification to a customization package.
4. Publish the customization package.

   For more information, see *Publishing or Unpublishing Customization Packages*.

Modifying the Windows Registry Parameters That Affect Local Synchronization

You can adjust the Windows registry parameters described in *Windows Registry Parameters That Affect Local Synchronization* by editing the platform_configuration.xml file, which is one of the metadata files in the customization package. For example, in the initialization_script- CDATA section, you can set the UpstreamSyncIsDefault parameter to make full synchronization the default synchronization type, as follows:

```
<initialization_script>
 <![CDATA[
 application.settings.set("UpstreamSyncIsDefault", 0);
 ]]>
</initialization_script>
```

For more information on synchronization, see *How Oracle CRM On Demand Desktop Synchronizes Data*

Preventing Data Loss When Upgrading a Customization Package

You can prevent data loss when upgrading a customization package by notifying your users on how to correctly manage customization packages.

In Oracle CRM On Demand Desktop, each role can have only one active customization package assigned to it. If you did not update the data structure in the customization package, then the customization package is compatible for the upgrade, and no new synchronization is needed.

However, if you did change the data structure in the customization package, then Oracle CRM On Demand Desktop might need to erase all the existing data in Microsoft Outlook, and to perform an initial synchronization. To prevent this, do the following:

1. Alert all Oracle CRM On Demand Desktop users in your company about the scheduled upgrade of the customization package.

2. Confirm the date on which the new package will be active.
3. Request that all Oracle CRM On Demand Desktop users in your company synchronize all changes before the scheduled upgrade date.

   When all users have synchronized, all changes are saved on the Oracle CRM On Demand server.
4. Add an end date to all currently active customization packages. This date must be equal to the scheduled upgrade date.

   You can do this by temporarily unpublishing the active customization packages. It is recommended that you perform this step during company downtime. This step prevents users receiving errors when there is no currently active published customization package. For more information, see *Publishing or Unpublishing Customization Packages*.
5. Make the new installer available to all the users in the company by uploading the new installer to the required location.
6. When Oracle CRM On Demand Desktop users log in after the scheduled date, the new customization package is available in the new installer which is active and ready for implementation.

   For more information, see *Overview of Installing Oracle CRM On Demand Desktop*. When Oracle CRM On Demand Desktop users log in, the new customization package is active and ready for use.

Using Filters in Oracle CRM On Demand Desktop

You can use filters in customization packages in Oracle CRM On Demand Desktop to limit what users can synchronize. By limiting synchronization, you can improve the performance of Oracle CRM On Demand Desktop.

The Oracle CRM On Demand Desktop Estimates page in the Oracle CRM On Demand Desktop setup wizard allows users to view an estimate of how many records they will synchronize with Microsoft Outlook.

Because Oracle CRM On Demand Desktop Estimates page estimates only the parent records, it is recommended that if you configure a preset filter, then you must run a test on some users and make sure that Oracle CRM On Demand Desktop returns approximately 2000 records from the estimation. For more information, see *Customizing Synchronization*.

Note: It is recommended to have 5000 records or less in Oracle CRM On Demand Desktop.

Configuring the Pin Mode for Records

The pin mode feature in Oracle CRM On Demand Desktop enables you to temporarily keep synchronized Oracle CRM On Demand Desktop records in Microsoft Outlook. Oracle CRM On Demand Desktop records that are kept in Microsoft Outlook can stay in Microsoft Outlook even if the records do not match your existing filters. For more information on filters, see *Filters Reduce the Number of Oracle CRM On Demand Records That Are Available in Oracle CRM On Demand Desktop*. By design, Oracle CRM On Demand Desktop records are automatically removed from Microsoft Outlook after a predefined time period.

You can use the Control Panel to configure the pin mode for records and do the following:

- Specify the amount of time that a record is kept in Microsoft Outlook
- Override the configured time period to manually remove a record from Microsoft Outlook

To configure the pin mode for records

1. Navigate to the Control Panel, and click Synchronization Exceptions.
2. In the Synchronizations screen, select Additions.
3. In the Additions screen, highlight the record that you want to configure, and then select Set PIN Mode.

   The Set PIN Mode feature sets the amount of time that synchronized records that are not filtered are displayed as records in Microsoft Outlook. The options include:

   - Unpin
   - For 1 day
   - For 1 week
   - For 1 month
   - For 1 year
   - Always
4. Click Close.

Configuring Online Lookup

Online lookup allows you to search and view all records visible to you in the Oracle CRM On Demand Desktop database while you are connected to your network. Online lookup can be invoked from any SalesBook button, as long as the user is online. Visibility is also defined by the access settings defined for the role. For more information on roles and profiles, see ***Oracle CRM On Demand Online Help*** .

While using online lookup, you can also keep records in Microsoft Outlook using the pin mode for records feature. For more information on using this feature, see *Configuring the Pin Mode for Records*.

Administering Log Files and Log Profiles

This topic describes how to administer log files and log profiles. It includes the following information:

- *Log Files and Dump Files You Can Use with Oracle CRM On Demand Desktop*
- *Assigning a Logging Profile to Oracle CRM On Demand Desktop*
- *Creating a Custom Logging Profile*

Log Files and Dump Files You Can Use with Oracle CRM On Demand Desktop

The following table describes log files and dump files that you can use with Oracle CRM On Demand Desktop. For information about how to enable these log files and dump files, see *Assigning a Logging Profile to Oracle CRM On Demand Desktop*.

Description	Where Stored
General Log. Includes information about general application events.	Windows Vista and Windows 7: **`C:\Users\user\AppData\Roaming\Oracle\CRMDesktop\Profile \Logs\GeneralLog`** Oracle CRM On Demand Desktop stores the first general log file as log.0000.txt. If this first file reaches 10 MB, then it creates a new file and increments the name of this file by 1, for example, log.0001.txt. A maximum of eight files can exist. If the eighth file reaches 10 MB, then Oracle CRM On Demand Desktop deletes the oldest General Log. **Note:** Log rotation does not occur until Microsoft Outlook is restarted. Files larger than 10 MB can be created. However, when Microsoft Outlook is restarted and the log file size is more than 10 MB, then Oracle CRM On Demand Desktop creates an additional log file.
Exception Log. Includes information about Oracle CRM On Demand Desktop exceptions. These log files are for Oracle internal use only.	Windows Vista and Windows 7: **`C:\Users\user\AppData\Roaming\Oracle\CRMDesktop\Profile \Logs\ExceptionLog`** Oracle CRM On Demand Desktop stores the first exception log file as ex_trace.0000.txt. If this first file reaches 10 MB, then it creates a file and increments the name of this file by 1, for example, ex_trace.0001.txt. A maximum of eight files can exist. If the eighth file reaches 10 MB, then Oracle CRM On Demand Desktop deletes the oldest file.
Crash Dump. Includes information about Outlook and Oracle CRM On Demand Desktop application failures.	Windows Vista and Windows 7: **`C:\Users\user\AppData\Roaming\Oracle\CRMDesktop\Profile \Logs\CrashDump`** A maximum of 48 crash dump files can exist. If 48 files exist, and if Oracle CRM On Demand Desktop must create another crash dump, then it deletes the oldest file.
SOAP Dump. Includes information about requests that Oracle CRM On Demand Desktop sends to the On Demand Server and replies that it receives from the On Demand Server.	Windows Vista and Windows 7: **`C:\Users\user\AppData\Roaming\Oracle\CRMDesktop\Profile \Logs\SoapDump`** A maximum of 48 SOAP dump files can exist. If 48 SOAP dump files exist, and if Oracle CRM On Demand Desktop must create another SOAP dump file, then it deletes the oldest SOAP dump.
Synchronization Dump. Includes synchronization events.	Windows Vista and Windows 7: **`C:\Users\user\AppData\Roaming\Oracle\CRMDesktop\Profile \Logs\ SyncDump`** A maximum of 48 synchronization dump files can exist. If 48 files exist, and if Oracle CRM On Demand Desktop must create another file, then it deletes the oldest file.

Assigning a Logging Profile to Oracle CRM On Demand Desktop

A logging profile is a set of parameters that determine logging settings. Oracle CRM On Demand Desktop comes with three predefined logging profiles and one custom profile. You can assign a predefined logging profile. You cannot change the set of parameters that a predefined logging profile uses. You can assign only one logging profile at a time.

To assign a logging profile to Oracle CRM On Demand Desktop

1. On the client computer, right-click the Oracle CRM On Demand Desktop icon in the system tray.
2. Choose Options and then click the Advanced tab in the Oracle CRM On Demand - Options dialog box.
3. Click Configure Logging and Reporting.
4. In the Logging Configuration dialog box, choose the Logging Policy using values from the following table.

Logging Policy	Log or Dump Enabled
Basic	General Log Oracle CRM On Demand Desktop sets the verbosity for the General Log to Info.
Detailed	Oracle CRM On Demand Desktop enables the following log files and dump files: ○ General Log ○ Exception Log ○ Crash Dump ○ Sync Dump ○ SOAP Dump Oracle CRM On Demand Desktop sets the verbosity for the General Log to the following: ○ Info for Detailed
Exhaustive	Oracle CRM On Demand Desktop enables the following log files and dump files: ○ General Log ○ Exception Log ○ Crash Dump ○ Sync Dump ○ SOAP Dump Oracle CRM On Demand Desktop sets the verbosity for the General Log to the following: ○ Debug for Exhaustive
Custom	Oracle CRM On Demand Desktop uses a custom profile that you define. For more information, see *Creating a Custom Logging Profile*.

5. Set the level of detail that Oracle CRM On Demand Desktop writes to the General Log. Under Logging Verbosity, select one of the following:
   - **Debug.** Logs information messages, warnings, and all errors.
   - **Info.** Logs only information messages.
   - **Warning.** Logs only warnings.
   - **Error.** Logs only errors.
   - **Fatal.** Logs only fatal errors.
6. Optional. Choose one or more of the following items:

- Log Application Exceptions
- Log Application Crashes
- Log Sync Dumps
- Log SOAP Dumps

At run-time, Oracle CRM On Demand Desktop creates log entries for each item you choose. For more information, see *Log Files and Dump Files You Can Use with Oracle CRM On Demand Desktop*.

7. Optional. Choose View Files for any item.

   If you choose View Files, then Oracle CRM On Demand displays the folder that includes the log files for the corresponding log type.

8. Click OK.

   Oracle CRM On Demand Desktop replaces the current logging settings with the setting that the profile you choose contains. It copies the parameters of this logging profile to the Oracle CRM On Demand Desktop logging settings in the Windows Registry in the following key:

   ```
   HKEY_CURRENT_USER\Software\Oracle\CRM On Demand Desktop\Logging
   ```

Creating a Custom Logging Profile

The example in this topic describes how to create a logging profile that enables General Log, Synchronization Dump, and SOAP Dump log entries and sets verbosity for the General Log to Warning.

To create a custom logging profile

1. In Microsoft Windows, choose Start and then click Run.
2. In the Run dialog box, enter REGEDIT and then click OK.
3. Locate the following key:

   ```
   HKEY_CURRENT_USER\Software\Oracle\CRM On Demand Desktop\LoggingProfiles
   ```

4. Create a subkey in the key that you located in Step 3.

   Enter a name for this subkey. At run time, Oracle CRM On Demand Desktop displays this name in the logging profile in the Logging Policy drop-down list of the Logging Configuration dialog box. You cannot localize a profile name. Oracle CRM On Demand Desktop stores this name in the registry with the settings that you specify for the profile. For more information, see *Assigning a Logging Profile to Oracle CRM On Demand Desktop*.

5. Create the following subkeys in the subkey that you create in Step 4. Create one subkey for each row that the following table contains.

Subkey Name	Enable	Description
GeneralLog	Yes	Set the value of the log_level parameter to the following decimal value: 3000 This value configures Oracle CRM On Demand Desktop to log only warnings. For performance reasons, it is recommended that Oracle CRM On Demand Desktop log only warnings during

Subkey Name	Enable	Description
		normal operations. For more information, see *Windows Registry Parameters That You Can Use with the General Log File*.
ExceptionLog	No	For more information, see *Windows Registry Parameters That You Can Use with the Exception Log File*.
CrashDump	No	For more information, see *Windows Registry Parameters That You Can Use with the Crash Dump File*.
SoapDump	Yes	For more information, see *Windows Registry Parameters That You Can Use with the SOAP Dump File*.
SyncDump	Yes	For more information, see *Windows Registry Parameters That You Can Use with the Synchronization Dump File*.

If you remove Oracle CRM On Demand Desktop at some point in the future, then it cleans all registry settings during removal.

6. Assign your custom profile.

   For more information, see *Assigning a Logging Profile to Oracle CRM On Demand Desktop*.

7. Optional. To distribute registry modifications in the network, you can use one of following:
   - Imported registration (.reg) files
   - regini.exe
   - Group policy
   - System policy

Configuring Web Mashups Controls

A mashup describes a Web application that combines multiple services into a single application. Any Web page can be interactively exposed through Microsoft Outlook’s form using Oracle CRM On Demand Desktop controls. The example in this procedure shows how to display a Google Maps location in an Account form that is driven by an address record in the Account form.

To configure Web mashup controls

1. Add the Web page control to the Account form by doing the following:
   a. Use an XML editor to open the forms_xx.xml file. Add cross reference here.
   b. Navigate to the SBL Account form.
   c. Add the following:

```
<web_page id="web_page_view">
 <url>http://www.maps.google.com/</url>
</web_page>
```

2. In the forms.js file, find the following:

   var current_form = new forms.account_form(forms.create_form_ctx(ctx));

3. Create a script and add below the line found in Step 2:

```
function GetPrimaryAddressURL()
{
 var params = [];
 var url = "http://maps.google.com/?";

 if(ctx.form.item["Primary Address Id"] != null)

 {
 var item = ctx.session.open_item(ctx.form.item["Primary Address Id"]);
 if(item != null)
 {
 if((item["Street Address"]).length > 0)
 params.push( (item["Street Address"]).replace(" ", "+") );
 if((item["City"]).length > 0)
 params.push( (item["City"]).replace(" ", "+") );
 if((item["Country"]).length > 0)
 params.push( (item["Country"]).replace(" ", "+") );
 if((item["State"]).length > 0)
 params.push( (item["State"]).replace(" ", "+") );
 }
 }

 if(params.length > 0) url += "&q=" + params.join(",+");
 else url += "&z=1";
 return url;
}
```

4. Add the following to the forms.js file, in the Account Controls section:

```
ctx.form.web_page_view.navigate = GetPrimaryAddressURL();
```

10 Customizing Oracle CRM On Demand Desktop

Customizing Oracle CRM On Demand Desktop

This chapter describes how to customize an implementation of Oracle CRM On Demand Desktop. It includes the following topics:

- *Customizing Oracle CRM On Demand Desktop*
- *Performing Typical Customization Work*
- *Process of Customizing Objects in Oracle CRM On Demand Desktop*
- *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*
- *Process of Adding a Predefined Oracle CRM On Demand Picklist to Oracle CRM On Demand Desktop*
- *Roadmap for Adding an MVG Field*
- *Setting Up Books for a Customized Package from Version 2.0*

Note: For additional information on the customization of Oracle CRM On Demand Desktop including how fields and other objects appear and behave in the user interface, and to set up book support for object types, see *Oracle CRM On Demand Desktop Customization Guide* .

Customizing Oracle CRM On Demand Desktop

This topic describes the options for modifying the customization package. It includes the following information:

- *Overview of Customizing Oracle CRM On Demand Desktop*
- *Files in the Customization Package*
- *Customizing How Oracle CRM On Demand Desktop Maps Fields*
- *Specifying Which Data Oracle CRM On Demand Desktop Removes from Microsoft Outlook*
- *Customizing Synchronization*
- *Customizing a Form*
- *Customizing a Toolbar*
- *Customizing a Dialog Box*
- *Customizing a View*
- *Customizing the SalesBook Control*
- *Customizing Meta Information*
- *Customizing Localization*
- *Customizing the URL Protocol to Use HTTPS*
- *Customizing the Email Address of the Support Team*

CAUTION: XML files and JavaScript files contain predefined configuration information that is critical to Oracle CRM On Demand Desktop operations. You must be careful if you modify any XML or JavaScript file. Make only the minimal modifications that you require. It is recommended that you unit test each change you make when possible.

For more information, see *Metadata That Describes the Oracle CRM On Demand Desktop Application*.

Overview of Customizing Oracle CRM On Demand Desktop

You can customize Oracle CRM On Demand Desktop to achieve the following:

- Extend the set of data that is available to the user.
- Add custom business logic to support the work that the user performs.

Oracle CRM On Demand Desktop can synchronize this information with Oracle CRM On Demand data in Microsoft Outlook. You can also make the data available to support offline usage when the user possesses limited or no connection to the Internet. You can create an interface where Oracle CRM On Demand Desktop stores the information the user requires to complete a business process. This technique allows the user to work in a single application rather than having to navigate between multiple applications, which simplifies the work process.

You Can Customize How Oracle CRM On Demand Desktop Processes Objects

To add custom logic for object processing in Microsoft Outlook, you can use Oracle CRM On Demand Desktop functionality, such as deduplication or data validation. You can also use JavaScript to create your own custom logic. Because you can use JavaScript to create, delete, modify, and convert objects, you can migrate your Oracle CRM On Demand Desktop data to Microsoft Outlook, including logic that supports a business process.

You can use a standard Oracle CRM On Demand Desktop form to display Oracle CRM On Demand objects, or you can create a completely new and custom form. A custom form can contain a native Microsoft Outlook control, such as a text box, or new controls that you develop for Oracle CRM On Demand Desktop, such as a lookup or a multiselect list. You can implement these forms through a custom toolbar, which can contain custom actions that you use JavaScript to implement. Some of the basic functionality that allows the user to use Oracle CRM On Demand Desktop functionality is predefined. The user can use Microsoft Outlook view controls on custom forms to display relationships between Oracle CRM On Demand objects. These views are configurable and support the functionality of native Microsoft Outlook views. You can also set the default views that Oracle CRM On Demand Desktop uses to display Microsoft Outlook objects.

Files in the Customization Package

To customize certain Oracle CRM On Demand Desktop features, you can modify XML files and JavaScript files in the customization package. Oracle CRM On Demand Desktop includes the following basic customization capabilities:

- Adjusting the business logic to suit the business environment
- Customizing the user interface
- Specifying security and data validation rules

For more information, see *Changes That Oracle CRM On Demand Desktop Makes During an Installation*.

About the Customization Package

The customization package consists of two types of files:

- Common platform JavaScript files that should not be changed during customization
- Product-specific files that include Javascript files and XML files.

You must tune product-specific files during customization to change the Oracle CRM On Demand Desktop behavior according to desired business logic in Oracle CRM On Demand.

Common platform files are added or replaced as required during an upgrade to the current version of Oracle CRM On Demand Desktop. For more details, see the table in the following topic and the table in the topic *XML Files in the Customization Package*.

Note: Make sure that you have not changed any of the common platform files during the customization process. If you have changed any of those files, then make sure to migrate all the changes manually. It is preferable to restructure your customization to move all the changes to product-specific files and avoid any changes to the common platform JavaScript files.

JavaScript Files in the Customization Package

The following table describes the JavaScript files that Oracle CRM On Demand Desktop includes in the customization package.

JavaScript File Name	Description
Common Platform Files **Note:** Do not change common platform files during customization.	
actions_support.js	Defines the common toolbar action support functions that are called from the actions.js file.
activity_processor.js	New for Version 2.0. Defines the common business logic for all the complex activity-related processing for calendar events, tasks, and email.
autoresolve_helpers.js	New for Version 2.0. Defines common helpers that are called from the autoresolver.js file.
data_model.js	Defines the common logic that handles the complex relations between objects and joined fields. This data model is initialized in the business_logic.js file.
form_helpers.js	Defines common JavaScript helpers for handling user interface events.
helpers.js	Defines the common utility functions to reuse in different files.
idle.js	Defines the Idle Processing Manager and common idle handlers.
md5.js	Defines how to implement message digest algorithm (MDA).

JavaScript File Name	Description
	Note: The md5.js file pertains to one of the many message digest algorithms available.
mvg_dialogs.js	Defines controls for multi-value groups (MVG).
raw_item_functions.js	Defines general functions that access Microsoft Outlook items.
od_helpers	Defines common utility functions to reuse in different files that are specific to Oracle CRM On Demand.
security_manager.js	Defines the common security model and manager used in the security_utils.js file.
Product-Specific Files	
actions.js	Defines the handlers for toolbar actions. This file includes functions for sharing email replies or forwards, and the Save Correspondence manager handler.
application_script.js	Defines the main entry point script file for the product and initializes the basic script context. This file includes the Scriptable First Run Assistant and supports extended activity processing (partially extracted to the activity_processor.js common file). It also includes two additional handlers for always sharing email replies or forwards and the Save Correspondence manager handler. Scriptable objects open from the Microsoft Windows Control Panel on the Sync Issues and Collision pages.
autoresolver.js	Defines functions with predefined rules to resolve conflicts when possible, based on business requirements. The major common code duplications reused in the file have been extracted to the new autoresolve_helpers.js common file to minimize and clarify the coding. This file implements a scriptable associations auto-resolver.
business_logic.js	Defines the business logic for product behavior and defines the basic data model for object relations. This file includes a flexible activity processing model with field length restrictions support. Options for email attachments processing have moved to the Options page. Search criteria have been updated for like type searches. This file supports opening association objects from the Microsoft Windows Control Panel. The file contains settings for the custom scripted SalesBook form.
forms.js	Defines the user interface actions for all the custom forms and dialogs. This file includes online lookups for SalesBook forms and supports record pin mode on the form. In addition, it supports custom view controls for external data storage and supports default visibility for subforms. This file validates addresses migrated from Microsoft Outlook contacts to Oracle CRM On Demand after installation.
security_utils.js	Defines the security definitions for types of objects and their relations.

JavaScript File Name	Description
	This file supports record pin mode. It restricts the deletion of contacts from Oracle CRM On Demand (either by adding the record to an exclusion list so that it is not included on the client even though it is present on the server, or by cancelling the deletion). This file contains additional restrictions for Opportunity.Product (where Product is a child record of the Opportunity record).

XML Files in the Customization Package

The following table describes the XML files that Oracle CRM On Demand Desktop includes in the customization package. For more information, see *Creating and Publishing the Customization Package*.

Note: All XML files in the customization package are considered product-specific, so you can customize these files as required for your business requirements.

XML File Name	Description
connector_configuration.xml	This XML file provides the following capabilities: • Defines objects that are synchronized • Defines the criteria that Oracle CRM On Demand Desktop uses to detect duplicate objects in the Oracle CRM On Demand database • Defines the preset filters for a custom synchronization For more information, see *Customizing Oracle CRM On Demand Desktop*. This file supports sliding dates that reflect the concept of recurrent activity handling. The sliding dates refer to the relative date filter criteria, for example: the next month (next 30 days), the last month (last 30 days), tomorrow, yesterday, and so on. Users can choose to only include activity records due in the next 30 days. This file also includes delete prohibit restrictions at the synchronizer level.
data_sources.xml	New for Version 2.0. Describes the data sources that are used for joining fields between objects and used in online lookups.
dialogs.xml	Defines the layout of a custom dialog box. For more information, see *Customizing a Dialog Box*. This file contains custom view controls that support an external database. The notes design uses a number of additional fields.
forms_XX.xml	These XML files provide the following capabilities: • Indicates the fields that Oracle CRM On Demand Desktop uses to store references between objects • Defines the form layout for each object • Defines the field validation rules on a form • Defines business logic in JavaScript • Defines controls that Oracle CRM On Demand Desktop uses on a form The business_logic.js file describes most business logic. It is recommended that you use JavaScript in one of the forms_xx.xml files to define business logic only if Oracle CRM On Demand Desktop executes this logic in the current form. If many objects are involved, then it is recommended that you define business logic in the business_logic.js file. **Note:** Throughout this document, the term forms_xx.xml refers generically to the forms_12.xml file. For more information, see *Customizing a Form*. This file contains custom view controls, combo boxes, and primary selectors. The file also contains auto-complete controls that support an external database. It supports an inline (an email To: line style) multi-value group (MVG).
info.xml	This XML file provides the following capabilities: • Defines the product name • Defines the version of the package • Defines compatible product and package versions • Defines the general comments for the package Oracle CRM On Demand Desktop does not currently use this information. You can use it to track the package version. You can use the product name and version to check compatibility.

Customizing How Oracle CRM On Demand Desktop Maps Fields

The od_basic_mapping.xml file describes objects you can add to Microsoft Outlook. These objects can be stored in either Microsoft Outlook storage (in .PST or .OST files) or in Oracle CRM On Demand Desktop's DB FACADE storage. It is a common practice to add objects displayed on the left side list of Outlook Explorer to Outlook storage, and to add other objects including associations and picklists to DB FACADE storage. Objects placed in Outlook storage are typically required to be displayed using Microsoft Outlook forms. Objects in DB FACADE storage are usually represented using dialogs.

The od_basic_mapping.xml file also defines the mapping between an Oracle CRM On Demand field and a Microsoft Outlook field. Using this file, you can extend the set of fields that Microsoft Outlook objects reference. Each description for an object defined in Microsoft Outlook storage includes the following information:

- The form that Oracle CRM On Demand Desktop uses to display the object. The form must be described in the forms_xx.xml file. The forms_xx file is dependent on the od_basic_mapping file. However, the form ID for each object is defined in od_basic_mapping.xml.
- The icons that Oracle CRM On Demand Desktop uses to display the object. Oracle CRM On Demand Desktop uses these icons in Microsoft Outlook views.
- The folder name for the object in the Microsoft Outlook navigation pane.
- A set of custom Microsoft Outlook views that Oracle CRM On Demand Desktop applies to the Microsoft Outlook view.
- Each description for an object defined in DB FACADE storage includes the object name and the icons that Oracle CRM On Demand Desktop uses to display the object. For more information, see *XML Code to Customize the Data That Oracle CRM On Demand Desktop Deletes If You Remove Oracle CRM On Demand Desktop*.

Specifying Which Data Oracle CRM On Demand Desktop Removes from Microsoft Outlook

The platform_configuration.xml file allows you to specify which custom data Oracle CRM On Demand Desktop removes from Microsoft Outlook. If the user uninstalls Oracle CRM On Demand Desktop or changes credentials, then Oracle CRM On Demand Desktop removes all the custom data from Microsoft Outlook that the od_basic_mapping.xml file describes. You can configure Oracle CRM On Demand Desktop to not delete data for a certain object type. Note that Oracle CRM On Demand Desktop does not remove Microsoft Outlook data, which is data that the user creates in the native Microsoft Outlook application. For more information, see *XML Code to Customize the Data That Oracle CRM On Demand Desktop Deletes If You Remove Oracle CRM On Demand Desktop*.

To specify which data Oracle CRM On Demand Desktop removes

1. To configure Oracle CRM On Demand Desktop not to delete data for a specific object type, define a rule named script for the appropriate type element in the platform_configuration.xml file.

   For example:

   ```
   type id="Action" rule="skip"/>
   ```

   In this example, Oracle CRM On Demand Desktop does not delete any data that is associated with the Action object type.

2. To configure Oracle CRM On Demand Desktop to conditionally not delete data for a specific object type, define an attribute named language for the rule you defined in Step 1.

   The language attribute defines the script language in the CDATA section. Oracle CRM On Demand Desktop supports only JScript, for example:

```
<type id="Action" rule="script" language="JScript">
 <![CDATA[ … JavaScript code … ]]>
</type>
```

3. To configure Oracle CRM On Demand Desktop not to delete data for multiple object types, add a separate rule for each object type, for example:

```
<type id="Action" rule="skip"/>
<type id="Opportunity" rule="skip"/>
```

Customizing Synchronization

The connector_configuration.xml file defines the objects that Oracle CRM On Demand Desktop synchronizes. It includes the synchronization settings that affect deduplication and preset filters. The connector_configuration.xml file provides the following capabilities:

- **Deduplication.** To determine whether an item already exists in the Oracle CRM On Demand database when Oracle CRM On Demand Desktop uploads the item from Microsoft Outlook to the Oracle CRM On Demand database, Oracle CRM On Demand Desktop uses special criteria that the connector_configuration.xml file describes. For more information, see *How Oracle CRM On Demand Desktop Avoids Duplicate Data*.
- **Preset filters.** Defines the preset filters for a custom synchronization. These preset filters help the user to synchronize only the data that the user requires, avoiding downloading data that the user does not require. You can configure any filter preset as the default. For more information on preset filters, see *Factors That Determine What Data an Oracle CRM On Demand Desktop User Can Access*.

For more information, see *XML Code to Customize Synchronization*.

Customizing a Form

The forms_12.xml file contains a definition for customized forms in Oracle CRM On Demand Desktop. These files allow you to customize forms, remove fields, change field names, and set custom list views.

The Microsoft Outlook version on which Oracle CRM On Demand Desktop runs determines which form Oracle CRM On Demand Desktop uses:

Forms for Microsoft Outlook 2007 and above are described in forms_12.xml.

Each form is described in a form element. Oracle CRM On Demand Desktop includes the following basic customization capabilities for these files:

- Changing field labels.
- Removing fields from the form that are not applicable to the work environment.
- Identifying fields on forms.
- Designating field type. Oracle CRM On Demand Desktop supports the following:

    - Native Microsoft Outlook field types, such as text box or check box
    - Custom Oracle CRM On Demand Desktop controls, such as currency control, lookup, and multiselect list

- Positioning a field on the form.
- Modifying custom Oracle CRM On Demand Desktop controls.
- Creating entirely new forms that use the look-and-feel of Microsoft Outlook. To enter data in these forms, Oracle CRM On Demand Desktop uses fields that the user defines.
- Programmatically applying custom behavior and logic to a form or field.
- Setting Microsoft Outlook view controls on a custom form to display relationships between Oracle CRM On Demand objects.
- Modifying interface elements, such as text in a system message, a dialog box, a label, or a caption.

For more information, see *XML Code to Customize Forms*, and *Correct Usage of the Forms_xx.XML File and Object ID*.

Validation Rules That Oracle CRM On Demand Desktop Uses When Customizing a Form

The forms.js file contains a description of the validation rules that Oracle CRM On Demand Desktop uses for the object that it displays on any form. You can configure Oracle CRM On Demand Desktop to do the following:

- Determine whether it enters information in the required fields
- Examine the format of any other field values, and so on
- Inform the user if a field validation failure occurs

Because validation uses JavaScript functions, you can combine these functions with JavaScript RegEx (JavaScript Regular Expressions) to configure a wide variety of format examination. For more information on defining validation rules, see *Defining the Validation Rules*.

Business Logic That Oracle CRM On Demand Desktop Uses with Customizing a Form

The forms.js files contain some of the business logic. You must implement them in JavaScript. JavaScript capabilities in Oracle CRM On Demand Desktop allow you to access any field of an object or any property of a Microsoft Outlook form. You can use this access to execute code on a specific event, such as opening a form, saving a form, and so on. You can configure Oracle CRM On Demand Desktop to fill fields automatically, format field values automatically, and disable or enable a control, depending on certain criteria.

Customizing a Toolbar

The toolbars12.xml and toolbars14.xml files describe the custom toolbars that Oracle CRM On Demand Desktop displays on a native Microsoft Outlook or custom form, or in a Microsoft Outlook window with programmable actions on the toolbar. Because some of the basic functionality is predefined, it is possible to use Oracle CRM On Demand Desktop without modification to meet your basic requirements. For more information, see *XML Code to Customize Toolbars*.

Customizing a Dialog Box

The dialogs.xml file defines the layout for a custom dialog box that Oracle CRM On Demand Desktop uses, such as the dialog box that it uses with an MVG, an address, or for email processing. Because the dialogs.xml file is an extension of the forms_12.xml file, it contains the same structure as the dialogs root element instead of the forms element in the forms_12.xml file. For more information, see *XML Code to Customize Toolbars*.

To customize the behavior of the MVG dialog box, you can modify the XML file that Oracle CRM On Demand Desktop includes in the customization package. You can customize the following MVG behaviors:

- Objects that Oracle CRM On Demand Desktop uses to create an association in the MVG dialog box.
- Fields that Oracle CRM On Demand Desktop specifies for the association.
- Format of the fields that Oracle CRM On Demand Desktop specifies for the association.
- Format in which Oracle CRM On Demand Desktop displays the Primary record. The default format is position (name), for example, District Manager 1 (Wasaka Takuda). You can specify the fields to display and the order in which Oracle CRM On Demand Desktop displays them.
- Fields that represent the associations you create. Oracle CRM On Demand Desktop displays these fields in the Microsoft Outlook view. Oracle CRM On Demand Desktop can display only the attributes of the record. If Oracle CRM On Demand Desktop stores the User name in the position record, then it can display only the user name for the position.
- The user permissions. The customization package uses security validation rules to describe the user permissions that function in the MVG. For example, if the user is not the primary user, then the user cannot delete users from the collection because Oracle CRM On Demand Desktop turns off the delete button for any user who is not a primary user.
- How a lookup control behaves. When you search for a record, a lookup control searches through the File As field for all associated objects.
- Hide the details of the parent record, such as the Opportunity Name.
- Add or remove association attributes for the associated record.
- Use an OK button instead of the Save and Close icon.

Customizing a View

The views.xml file describes the view configurations that Oracle CRM On Demand Desktop uses on Oracle CRM On Demand Desktop forms and in Microsoft Outlook views. Each Microsoft Outlook view uses an xml file that is native to Microsoft Outlook. Oracle CRM On Demand Desktop reuses this XML description for custom views. For more information on customizing a view in Oracle CRM On Demand Desktop, see *XML Code to Customize Views Used for Microsoft Outlook Explorer List*.

Customizing the SalesBook Control

The lookup_view_defs.xml file sets configuration options for the SalesBook control in Microsoft Outlook. This control defines references between objects. Oracle CRM On Demand Desktop uses it primarily in lookup controls. You can use the lookup_view_defs.xml file to specify which objects Oracle CRM On Demand Desktop makes available through each SalesBook control. You can also set filters for the objects you must display in the SalesBook control. For more information, see *XML Code to Customize the Sales Book Control*.

Customizing Meta Information

The od_meta_info.xml file contains the following meta information:

- A description of the object types that Oracle CRM On Demand Desktop supports.
- Fields that are defined and the type for each field.
- XML element names that Oracle CRM On Demand Desktop uses to build or parse an Oracle CRM On Demand message. Oracle CRM On Demand Desktop uses information about the relations between objects from the file for the Oracle CRM On Demand message.
- The definition of each object that Oracle CRM On Demand Desktop supports contains a unique name, an XML element, and an XML collection element that Oracle CRM On Demand Desktop uses in an Oracle CRM On Demand message.

Every object field includes a name, an Oracle CRM On Demand data type, and an XML element. Oracle CRM On Demand Desktop uses this information in an Oracle CRM On Demand message to display values and filters for that field. For more information, see *XML Code for Meta Information*.

Customizing Localization

To customize localization, you can customize the resource files.

To customize localization

1. Add a new resource for the custom label and attribute name or warning message in the following files:
   - Use the package_res.xml file for the default resources.
   - Use the package_res.xx_YY.xml file for the specific locale.

   where:
   - xx_YY is the language you use in your implementation.

   For example, use package_res.pt_BR.xml for Portuguese Brazilian.
   The following standards determine the locale naming convention:
   - xx is the ISO 639-1 standard for the language.
   - YY is the ISO 3166-1 standard for the country. This standard supports dialects and language adoptions for specific countries.
2. Add the XML files to the customization package.
3. Republish the customization package.
   For more information, see *Publishing or Unpublishing Customization Packages*.

Customizing a Language for the Forms Files

To customize the behavior of the forms_xx.xml file that is predefined with Oracle CRM On Demand Desktop, you can use a forms file that is specific to a language. For example, forms_12.ja_JP.xml for JPN (Japanese).

To customize a language for the forms files

1. In Windows Explorer, navigate to the directory that contains the forms_xx.xml file.

   For more information, see *Files in the Customization Package*.
2. Right-click the forms_xx.xml file, and then choose the Copy menu item.
3. Rename this copy to indicate that it is a specific to a language, for example:

   ```
   forms_12.ja_JP.xml
   ```
4. Use an XML editor to open the file you renamed in Step 3, and then make any changes that are required to support the language.

   You can use this file to change the layout of the form, such as adding new fields. For example, in Japanese, you might define three different fields for the Account Name:

   - One field represents the native name of the company for the account.
   - One field represents the Kanji symbol for the company.
   - One field represents the phonetic name of the company.

   If you make these changes, then make sure you also perform the other configuration tasks that the changes require, such as defining field mappings and changing the form layout.
5. (Optional) To change text strings, repeat Step 2 and Step 3 but create a copy of the package_res.xml file, rename it, and then edit the text strings, for example:

   ```
   package_res.ja_JP.xml
   ```

Customizing the URL Protocol to Use HTTPS

You can customize the URL protocol to use HTTPS (Hypertext Transfer Protocol Secure).

To customize the URL protocol to use HTTPS

1. Open a DOS command line interface, and then type **regedit.exe**.
2. Set the Connector:Protocol registry parameter to https.

   For more information, see *Modifying the Windows Registry to Change Oracle CRM On Demand Desktop Behavior*.

Customizing the Email Address of the Support Team

The package_res.xml file defines various resources for the customization package. In this file, you can specify the email address of the support team to which the user sends feedback.

To customize the email address of the support team

- Modify the following code of the package_res.xml file:

  ```
  <!-- Feedback page -->
  ```

```
<str key="support_email">email_address</str>
```

where:

- email_address is the email address to which Oracle CRM On Demand Desktop sends requests for support

  For example:

  ```
  <str key="support_email">support@your_company.com</str>
  ```

  If you specify the email address in the support_email variable, and if the user clicks Send Feedback on the Feedback tab in the Options dialog box, then Oracle CRM On Demand Desktop opens a new email message. Oracle CRM On Demand Desktop automatically enters the value that you specify in the support_email variable. It enters this information in the address To line of this email message.

  If the user clicks the Send Feedback button on the Feedback tab in the Options dialog box, and if the email address is not specified, then Oracle CRM On Demand Desktop opens the email without an email address in the To line. A support email address is not specified in the predefined Oracle CRM On Demand Desktop product.

Performing Typical Customization Work

This topic describes other customization work that you might find useful. It includes the following information:

- *Customizing the Product Name*
- *Localizing Values*
- *Example of Displaying a Custom Oracle CRM On Demand Field in an Oracle CRM On Demand Desktop Form*
- *Making a Field in Oracle CRM On Demand Desktop Read-Only*

Customizing the Product Name

The Oracle CRM On Demand Desktop clients display the following text in several locations:

- Oracle CRM On Demand
- Oracle CRM On Demand Desktop
- Microsoft Outlook

You can change this text to a custom value.

To customize the product name

1. Use an XML editor to open the package_res.xml file.

   For more information, see *Files in the Customization Package*.

   Create or modify any of the following attributes, as required:

   - `<str key="app_name">Oracle CRM On Demand Desktop </str>`
   - `<str key="pim_name">Microsoft Outlook </str>`

   - `<str key="remote_app_name">Oracle CRM On Demand </str>`

   For example, in the remote_app_name attribute, change Oracle CRM On Demand to the name of your company.
2. Save your changes, republish, and then test your changes.
   For more information, see *Publishing or Unpublishing Customization Packages*.

Localizing Values

You can modify a single file to do the following work:

- Support different languages.
- Change a label value that is included in multiple locations in the user interface.

To localize values

1. Use an XML editor to open the package_res.xml file.
   For more information, see *Files in the Customization Package*.
2. To define a localizable string, add the following code to the package_res.xml file:
   <str key="string_id">localizable_string</str>
   where:
   - string_id is the Id of the localizable string
   - localizable_string is the localizable string
3. Use the localizable string Id in every location where Oracle CRM On Demand Desktop must display the string.
   You must use different formats to define the string in different types of files. Use values from the following table.

File Type	Description

Any XML file except for the views.xml file.	Use the following format:
	#string_id
	For example:
	`<cell size="22">`
	` <static id="account_label" tab_order="6">`
	` <text>#lbl_account</text>`
	` </static>`
	`</cell>`

The views.xml file.	Use the following format:
	$string_id$
	For example:
	`<str key="all_accounts">`
	`<![CDATA[`
	`<?xml version="1.0"?> <view`
	` type="table"> <viewname>$view_siebel_accounts$</viewname>`

File Type	Description

	`................................ </view>`
	`]]>`
	`</str`

Any JavaScript file.	Use the following format:
	session.res_string("string_id")
	For example:
	`ui.message_box(0, session.res_string("msg_general_error"), session.res_string("msg_general_error_caption"), 0x40);`

Example of Displaying a Custom Oracle CRM On Demand Field in an Oracle CRM On Demand Desktop Form

This topic gives one example of how to display a custom Oracle CRM On Demand field in an Oracle CRM On Demand Desktop form. You might use this feature differently, depending on your business model. In this example, you display the Mail Stop field on the Contact form in Oracle CRM On Demand Desktop. You make this field available through Oracle CRM On Demand and then customize Oracle CRM On Demand Desktop to synchronize and display the field. You modify the following files:

- od_meta_info.xml
- od_basic_mapping.xml
- forms_12.xml
- package_res.xml

For more information, see *Files in the Customization Package*.

To display a custom Oracle CRM On Demand field in an Oracle CRM On Demand Desktop form

1. Define the objects and fields to synchronize:
   a. Use an XML editor to open the od_meta_info.xml file.

      For more information, see *Files in the Customization Package*.
   b. In the od_meta_info.xml file, locate the following element:

      ```
      object TypeId='Contact'
      ```

      Several child field tags reside in the object TypeId='Contact' element. These children define the fields for the Contact object.
   c. Add the following field element as a child to the object TypeId='Contact' element:

      ```
      <field Name='Mail Stop' Label='Mail Stop' DataType='DTYPE_TEXT'
      IOElemName='MailStop' />
      ```
   d. Repeat Step b and Step c for the following element:

```
object TypeId='Account.Contact'
```

e. Repeat Step b and Step c for the following element:

```
object TypeId='Opportunity.Contact'
```

f. Save and close the od_meta_info.xml file.

2. Map the Mail Stop field from the Contact object in the Oracle CRM On Demand database to a field in Oracle CRM On Demand Desktop:

   a. Use an XML editor to open the od_basic_mapping.xml file.

      For more information, see *Files in the Customization Package*.

   b. In the od_basic_mapping.xml file, add a new field element to the type element, using values from the following table.

Element	Value
type id	Contact

   c. Add the following code to the element you created in Step b.

```
<field id="Mail Stop">
 <reader class="mapi_user">
 <user_field id="od Mail Stop" ol_field_type="1"></user_field>
 <convertor class="string"></convertor>
 </reader>
 <writer class="Microsoft Outlook_user">
 <user_field id="od Mail Stop" ol_field_type="1"></user_field>
 <convertor class="string"></convertor>
 </writer>
</field>
```

   d. Save and close the od_basic_mapping.xml file.

3. Insert a label and the Mail Stop field just below the Job Title field on the Contact form:

   a. Open the forms_12.xml file, and then locate the cell that contains the #lbl_job_title label control.

   b. Insert the following XML code immediately after the cell that contains the #lbl_job_title label control:

```
<cell size="22">
 <static id="lbl_MailStop" tab_order="6">
 <text>#lbl_mail_stop</text>
 </static>
</cell>
```

      The text #lbl_mail_stop specifies a key that the package_res.xml file uses to determine the localized value for the label.

   c. Locate the section that is labeled with the left side fields comment.

      This section resides in the element that resides in the form that contains the OD Contact ID.

   d. To add the text field control, insert the following XML code immediately above the cell that contains the ContactToAccount MVG control, and just below the cell that contains the status_image control:

```
<cell size="22">
 <edit id="MailStop" tab_order="7">
 <field value="string">Mail Stop</field>
 </edit>
</cell>
```

e. Locate the cell size element, and then change its value to the following:

```
<cell size="185">
```

To make room for the new field, you must increase the size of the cell that houses all of the child objects. In this example, you change the cell size from 155 to 185.

4. Add the following code to the package_res.xml file:

```
<str key="lbl_mail_stop">Mail stop:</str>
```

Add this code as a child of the res_root element under the `<!-Contact Form` comment.

This code provides localized values and images to the Oracle CRM On Demand Desktop client. This code allows the Contact form to render the Mail Stop label through a key value. The package_res.xml file provides localized values and images to Oracle CRM On Demand Desktop. Because you added the new Mail Stop field to the Contact form, you must provide the text for the label. When you modified the forms_12.xml file, you created a label control that contains #lbl_mail_stop for the text value. This control identifies the key to use in the package_res.xml file.

5. Republish the updated package files.

During the next synchronization, Oracle CRM On Demand Desktop uses the updated files to apply the modifications to the Contact form. The Mail Stop field is available on the Contact form and Oracle CRM On Demand Desktop synchronizes the values in this field with the Oracle CRM On Demand Desktop server. For more information, see *Publishing or Unpublishing Customization Packages*.

Making a Field in Oracle CRM On Demand Desktop Read-Only

This topic describes how to add a field in Oracle CRM On Demand Desktop and how to make the field read-only so it cannot be modified by users.

To make a field in Oracle CRM On Demand Desktop read-only

1. Use a JavaScript editor to open the forms.js file.
2. Locate the field that you want to make read-only.
3. Append the control.enabled attribute to the field, and set it to False, for example:

```
ctx.form.control_id.enabled = false;
```

4. Save and close the forms.js file.

Adding a Default Value to a Field in Oracle CRM On Demand Desktop

This topic describes how to configure Oracle CRM On Demand Desktop to add a default value to a field when the user creates a new record.

To add a default value to a field in Oracle CRM On Demand Desktop

1. Use a JavaScript editor to open the business_logic.js file.
2. Locate the following function:

   ```
   create_ondemand_meta_scheme2
   ```

3. Add the following string to the create_siebel_meta_scheme2 function:

   ```
   scheme.objects.get_object("Service_Request").get_field("Status")["initial_value
   "] = "Open"
   ```

   where:

   - Service_Request is the TypeId of the object. If you want to identify the TypeId, then you can locate the corresponding type description in the od_basic_mapping.xml file. The id attribute of the type element is the TypeID.
   - Status is the name of the field in which Oracle CRM On Demand Desktop adds a default value. If you want to identify the name of the field, then you can locate the corresponding type description and field element inside the type element in the od_basic_mapping.xml file. The id attribute of the corresponding field element is the name of the field.
4. Save the business_logic.js file, and then test your work.

Process of Customizing Objects in Oracle CRM On Demand Desktop

This topic gives one example of customizing objects in Oracle CRM On Demand Desktop. You might use this feature differently, depending on your business model.

This topic is a step in *Roadmap for Installing Oracle CRM On Demand Desktop*.

To customize objects in Oracle CRM On Demand Desktop, do the following:

1. *Defining the Custom Object*
2. *Defining Synchronization for a Custom Object*
3. *Adding Custom Views in Microsoft Outlook*
4. *Defining the User Interface*
5. *Defining the Validation Rules*
6. *Adding Custom Logic*
7. *Adding Custom Logic*
8. *Defining the Toolbar*
9. *Defining Other Options*

For more information on:

- Overview of XML files that you modify in this example, see *Customizing Oracle CRM On Demand Desktop*.
- Details about tags in XML files that you modify in this example, see *XML Files Reference*.
- Lengthy XML code that you use in this example, see *Additional Code in the Customization Example*.

Defining the Custom Object

When you add a new object to Microsoft Outlook, you describe the structure of the object, create mapping between fields, lists, and so on. You make these customizations in the od_basic_mapping.xml file.

When adding a custom object from Oracle CRM On Demand Desktop, enter the custom object names in the XML file using the following structure:

- For Custom Objects 1 -3, enter `Custom Object X` for the custom object name.

  Spaces are allowed for Custom Objects 1 -3 names.
- For Custom Objects 4-x, enter `CustomObjectX` for the custom object name.

  Spaces are allowed for Custom Objects 4-x (picklist object) names in some instances. For more information, see *Guidelines for Defining a List Object*.

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

To define the custom object

1. Use an XML editor to open the od_basic_mapping.xml file.

   For more information, see *Files in the Customization Package*.
2. To define the name of the custom object, add the following sample code to the od_basic_mapping.xml file and modify the values as required:

```
<type id="Activity" display_name="#obj_activity_plural" folder_type="10">
 <form message_class="IPM.Contact.OnDemand.Activity"
icon="type_image:Event:16" large_icon="type_image:Event:32"
display_name="Activity">OnDemand Activity</form>
</type>
```

   If the object is stored in DB FACADE storage, then add the following XML code:

```
<type id="Currency" icon="type_image:Currency:16"></type>
```

   For more information, see *Example Code for the Activity Form*.
3. Define a set of fields for the custom object. The fields that you define for this example are described in the following table.

Field Label	Field Name	Field Type
Description	Description	Text
Type	Type	Picklist
Priority	Priority	Picklist
Owner	Primary Owner Id	Lookup
Account	Account Id	Lookup

Field Label	Field Name	Field Type
Opportunity	Opportunity Id	Lookup
Contacts	No field on this object	MVG
Employee	No field on this object	MVG
Planned Start	Planned	datetime
Planned Completion	Planned Completion	datetime
Due	Due	datetime
Status	Status	Picklist
Comments	Comment	Textarea

For more information, see *Fields That Oracle CRM On Demand Desktop Uses for the Custom Object*. To examine the code you must add for this step, see *XML Code That Defines a Set of Custom Fields*.

4. Define intersection objects for many-to-many relationships.

   You do not define many-to-many relationships in Step 3. Because there is a many-to-many relationship between contacts and activities and between users and activities, you must define more objects that contain links to activity and contact, or activity and user. The remaining description for a many-to-many relationship is the same as for other objects where you specify the object name and object fields. This object can also contain a field that indicates if this intersection record is a primary or not a primary.

   To examine the code you must add for this step, see *XML Code That Defines a Set of Custom Fields*.

5. Define the lists.

   For more information, see *Guidelines for Defining a List Object*. To examine the code that you must add for this step, see *XML Code That Defines a List*.

Example Code for the Activity Form

The example code defines Activity as the form to display for this object. The form layout is defined later when you define the user interface. In this example, you add an Activity object to Microsoft Outlook. In Oracle CRM On Demand, this object is named Action. To add this object to Microsoft Outlook, you modify the od_basic_mapping.xml file. You add the type element to describe the new object. You can also use the display_name attribute of the type element to define the folder name for the object in Microsoft Outlook. You can use the folder_type attribute to define the native Microsoft Outlook object that is the base for the custom object folder type.

Fields That Oracle CRM On Demand Desktop Uses for the Custom Object

The od_basic_mapping.xml file describes each of the fields for the custom object. Oracle CRM On Demand Desktop maps each field to a custom field, except for the Description and Comments fields:

- Oracle CRM On Demand Desktop maps the Description field to the Last Name field, which is a native field in Microsoft Outlook.

- Oracle CRM On Demand Desktop maps the Comment field to the Body field because Oracle CRM On Demand Desktop does not support the textarea field. Therefore, Oracle CRM On Demand Desktop uses the native Microsoft Outlook control that displays the Body field value.

Guidelines for Defining a List Object

Because the custom object stores items that the user chooses in a list, you describe the list field in the same way as a string field. Therefore, you must describe the object that stores all list values. Each list uses a separate object to store the values of the list. You must build the IDs for these objects in the following order:

1. Object name, where the list is used, for example, Activity.
2. Name of the field containing the list name, for example, Type field on Activity object.
3. Name of the list, for example, Picklist.

Ensure the Type list on the Activity object includes the ID of the ActivityTypePicklist object.

Note: There must not be any spaces between the three values. The following naming is incorrect: Activity TypePicklist. However, if a space is present inside an object or field's name, for example, Activity Timing object, or Measure Type field name, then these spaces are allowed in the picklist's name. For example, the following list object name is correct: Activity TimingMeasure TypePicklist

Defining a List Object

Complete the following procedure to define a list object.

To define a list object

- You must define the following set of standard fields:
  - Label
  - Value (string)
  - SortOrder (integer)
  - IsDefault (bool)

Defining Synchronization for a Custom Object

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

This topic describes how you can define a custom object so that Oracle CRM On Demand Desktop can synchronize it with an Oracle CRM On Demand object.

To define synchronization for a custom object

1. Use an XML editor to open the connector_configuration.xml file.

   For more information, see *Files in the Customization Package*.
2. Define the Links section.

   The Links section contains a set of fields that reference other objects. You must define these references to allow the synchronization engine to synchronize objects in the correct order and to download the related items. Add the following XML code to define the links:

```
 <type id="Action">
 <view label="Activity" label_plural="Activities"
small_icon="type_image:Event:16" normal_icon="type_image:Event:24"
large_icon="type_image:Event:48"></view>
 <synchronizer name_format=":[:(Description):]">
 <links>
 <link>Account Id</link>
 <link>Opportunity Id</link>
 <link>Primary Owner Id</link>
 </links>
 </synchronizer>
 </type>
```

3. Define the deduplication keys.

   Because the business environment for this example requires that activities are the same if their descriptions are the same, one natural key is defined. That key is Description. Add the following XML code to the synchronizer element:

```
 <natural_keys>
 <natural_key>
 <field>Description</field>
 </natural_key>
</natural_keys>
```

4. Add the following descriptions to the connector_configuration.xml file for the intersection records that you defined for contacts and user in Step 4 in the topic *Defining the Custom Object*:

```
 <type id="Action.Employee.Association">
 <view label="Activity Employee" label_plural="Activity Employees"
small_icon="type_image:Generic:16" normal_icon="type_image:Generic:24"
large_icon="type_image:Generic:48" suppress_sync_ui="true"></view>
 <synchronizer name_format=":[:(UserName) :]">
 <links>
 <link>EmployeeId</link>
 <link>ActionId</link>
 </links>
 </synchronizer>
 </type>
 <type id="Action.Contact.Association">
 <view label="Activity Contact" label_plural="Activity Contacts "
small_icon="type_image:Generic:16" normal_icon="type_image:Generic:24"
large_icon="type_image:Generic:48" suppress_sync_ui="true"></view>
 <synchronizer name_format=":[:(ContactName) :]">
 <links>
 <link>ActionId</link>
 <link>ContactId</link>
 </links>
 </synchronizer>
 </type>
```

Adding Custom Views in Microsoft Outlook

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

After you define a folder in Microsoft Outlook that displays the custom object, you define the Microsoft Outlook views that apply to this folder.

To add custom views in Microsoft Outlook

1. Use an XML editor to open the od_basic_mapping.xml file.

   For more information, see *Files in the Customization Package*.

2. Add the following custom_views section to the Activity type element in the od_basic_mapping.xml file:

```
 <type id="Activity" display_name="#obj_activity_plural" folder_type="10">
 <form (…) >OnDemand Activity</form>
 <custom_views default_name="#view_od_activities">
 <view id="all_activities" name="#view_od_activities"></view>
 <view id="all_activities_by_owner"
name="#view_od_activities_by_primary_owner"></view>
 <view id="all_activities_by_priority"
name="#view_od_activities_by_priority"></view>
 </custom_views>
...
```

3. Make sure the views.xml file describes the views that you define in Step 2.

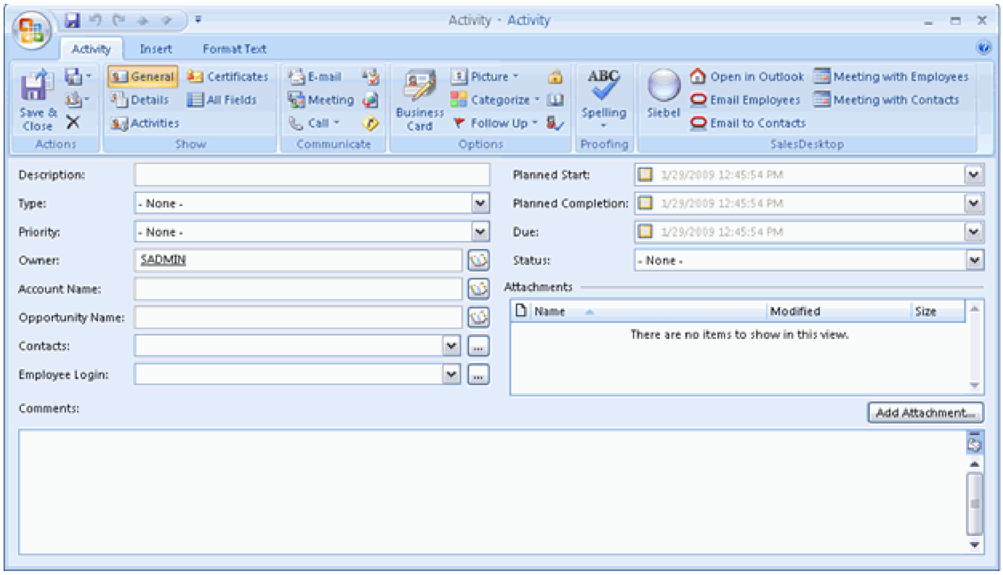
   For more information on views, see *Customizing a View*.

Defining the User Interface

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

In this topic, you define the user interface for the custom object. To provide the user with a way to work with the custom object, you must define the form that Oracle CRM On Demand Desktop uses to display the custom object. The custom object is configured to use the custom OnDemand Activity form in the od_basic_mapping.xml file that you modified in *Defining the Custom Object*. Therefore, you must create the OnDemand Activity form.

The following figure illustrates the layout of the Activity Form.



Description:
Type: - None -
Priority: - None -
Owner: SADMIN
Account Name:
Opportunity Name:
Contacts:
Employee Login:
Comments:

Planned Start: 1/29/2009 12:45:54 PM
Planned Completion: 1/29/2009 12:45:54 PM
Due: 1/29/2009 12:45:54 PM
Status: - None -

Attachments

Name	Modified	Size

There are no items to show in this view.

Add Attachment...

A set of *cells* describes the form layout. A cell can be empty, or it can contain a control or a stack. A single stack can contain numerous cells. Therefore, the form is divided into the following two parts:

- One part includes visible controls.
- One part includes hidden controls.

This example uses this technique because the example uses native Microsoft Outlook forms as a base for custom forms. Although you can modify the position of a native Microsoft Outlook control, you cannot remove it entirely from the form. Therefore, you move the unused controls that are native to Microsoft Outlook to a location where the user cannot view them, which is typically a small cell.

This form is a prototype that helps to visualize the cells, stacks of cells, and the order in which you use the cells and stacks. This visualization helps to reduce the wide range of combinations of cells and stacks that you can describe. The visualization is restricted to only those cells and stacks that you can support effectively. You can add new fields, remove fields, reorder fields, and apply any other changes during development and testing.

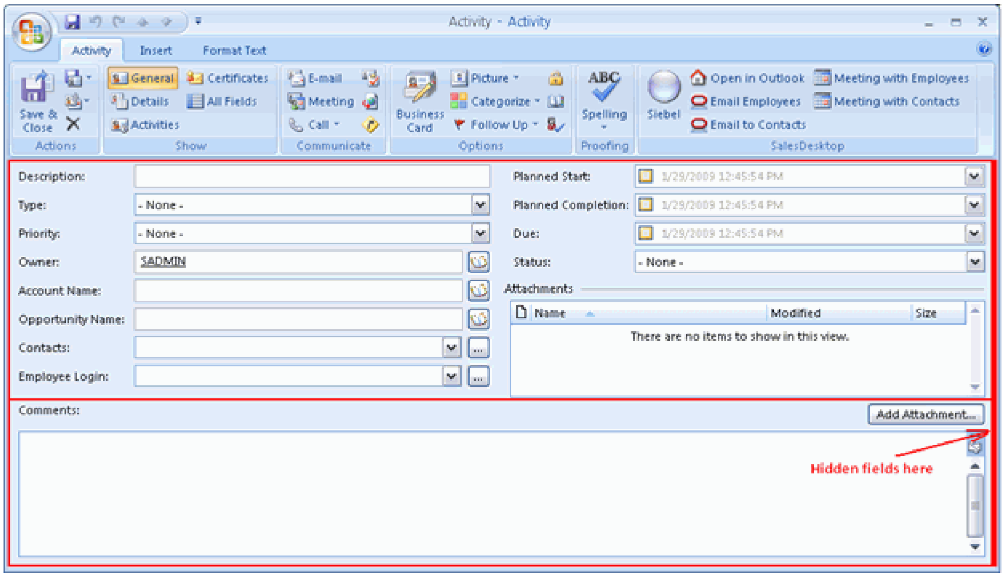
To define the user interface

1. Use an XML editor to open the forms_12.xml file.

   For more information, see *Files in the Customization Package*.
2. To divide the form into a visible section and a hidden section, add the following XML code to the forms_12.xml file:

```
<form id="OnDemand Activity">
<page id="General" tag="0x10A6" min_height="335" min_width="520">
<cell>
<stack layout="horz" padding="10">
<!-- visible section -->
<cell>
<!-visible fields here -->
</cell>
<!-- hidden section -->
<cell size="1">
<!-hidden fields here -->
</cell>
</stack>
</cell>
</page>
</form>
```

3. Divide the visible section into two vertical sections that include a top section and a bottom section, as illustrated in the following figure:



Description:	
Type:	- None -
Priority:	- None -
Owner:	SADMIN
Account Name:	
Opportunity Name:	
Contacts:	
Employee Login:	
Planned Start:	1/29/2009 12:45:54 PM
Planned Completion:	1/29/2009 12:45:54 PM
Due:	1/29/2009 12:45:54 PM
Status:	- None -

Attachments

There are no items to show in this view.

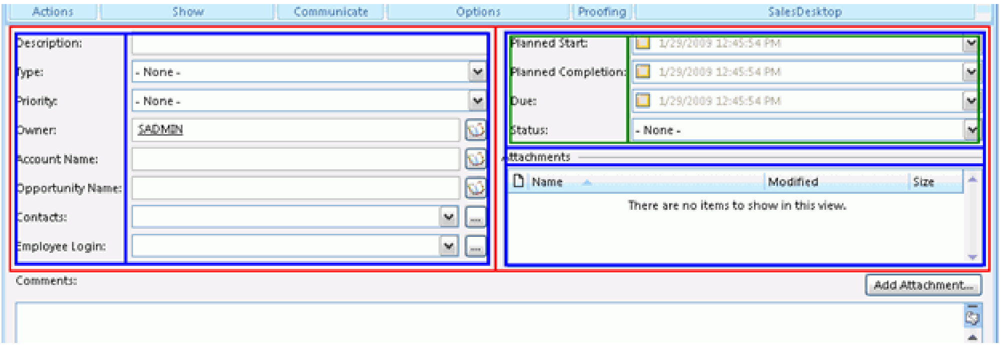
Comments:

Hidden fields here

Although not required, this step helps to support the current layout of the form. Add the following code:

```
<form id="OnDemand Activity">
(...............)
<!-- visible section -->
<cell>
<stack layout="vert" padding="5" spacing="5">
<!-top section-->
<cell size="220">
<!-top section fields here-->
</cell>
<!-bottom section-->
<cell>
<!-bottom section fields here-->
</cell>
</stack>
</cell>
(...............)
</form>
```

4. Divide the top section into two parts: one that displays a left column of fields and the other a right column of fields, see the following figure:



Description:	
Type:	- None -
Priority:	- None -
Owner:	SADMIN
Account Name:	
Opportunity Name:	
Contacts:	
Employee Login:	
Planned Start:	1/29/2009 12:45:54 PM
Planned Completion:	1/29/2009 12:45:54 PM
Due:	1/29/2009 12:45:54 PM
Status:	- None -

Attachments

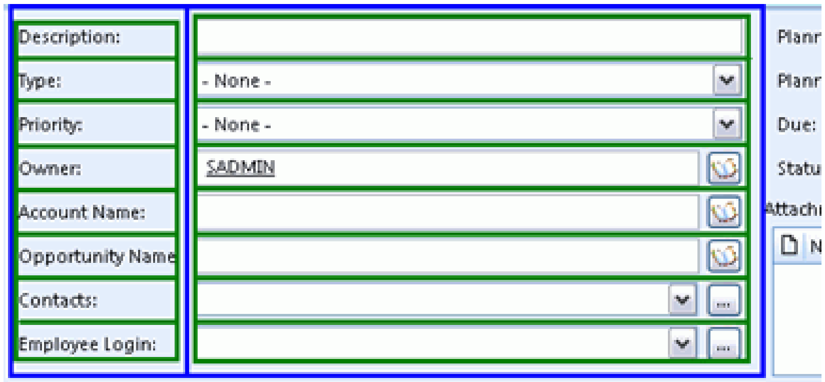
There are no items to show in this view.

Comments:

Add the following code:

```
<-- top section -->
<cell size=220>
 <stack layout="horz" spacing="5">
 <cell>
 <stack layout="horz" spacing="3">
 <!-- left side captions -->
 <cell size="105">
 </cell>
 <!-- left side fields -->
 <cell> </cell>
 </stack>
 </cell>
 <cell>
 <stack layout="vert" spacing="5">
 <cell size="105">
 <stack layout="horz" spacing="3">
 <!-- left side captions -->
 <cell size="110">
 </cell>
 <!-- left side fields -->
 <cell>
 </cell>
 </stack>
 </cell>
 <cell size="13">
 <!-attachments caption and separator here -->
 </cell>
 <cell>
 <!-attachments view here -->
 </cell>
 </stack>
 </cell>
 </stack>
</cell>
```

5. Divide the left column into cells where you can place captions for fields, see the following figure:



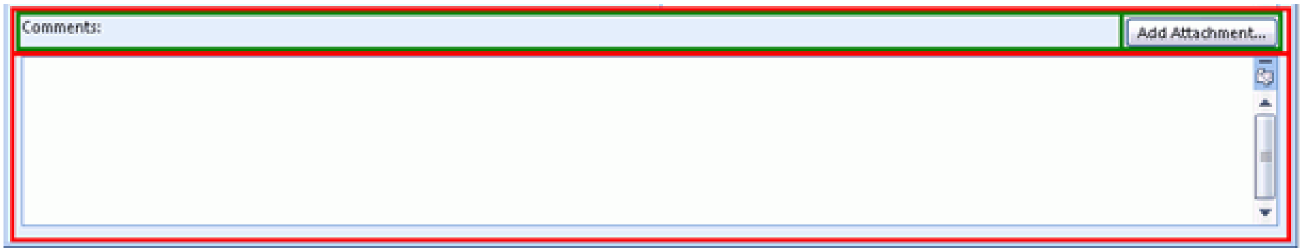
Description:
Type: - None -
Priority: - None -
Owner: SADMIN
Account Name:
Opportunity Name
Contacts:
Employee Login:

For more information, see *XML Code That Creates Cells*.

6. Divide the right column into cells where you can place fields.

Because the code that you use to complete this step is similar to the code you use in Step 5, you can copy and modify the code that is described in *XML Code That Creates Cells*.

7. Define the Comments section of the Activity form, as illustrated in the following figure:



The Comments section is divided into the following parts:

- Comments caption and the Add Attachment button
- Comments field

The Add attachment button is described separately. For more information on buttons, see *Adding a Label, Button, and Selector Control*.

Add the following code:

```
<!-- visible section -->
<cell>
<stack layout="vert" padding="5" spacing="5">
<!-top section-->
<cell size="220">
<!-top section fields here-->
</cell>
<!-bottom section-->
<cell size="22">
<stack layout="horz">
<cell>
<control id="0x20022" class="static" tab_order="30">
<text>Comments:</text>
</control>
</cell>
<cell size="110" attraction="far">
<control class="button" id="0x20059" tab_order="29"
on_click="LinkAttachment()">
<text>Add attachment…t</text>
</control>
</cell>
<cell size="5" attraction="far"></cell>
</stack>
</cell>
<cell>
<control id="0x103f" tab_order="31"></control>
</cell>
</stack>
</cell>
```

8. Make sure the description you create supports the version of Microsoft Outlook that your users currently use.

   For more information, see *Correct Usage of the Forms_xx.XML File and Object ID*.

Correct Usage of the Forms_xx.XML File and Object ID

Outlook 2007 and above use different IDs for native Microsoft Outlook controls. Because Oracle CRM On Demand Desktop supports these versions of Microsoft Outlook, you must create a different description of the form for each version that your implementation uses, where each form uses the same description of the layout, but the native Microsoft Outlook control IDs are different.

Because the code in this topic supports Microsoft Outlook 2007 and above, you must add it to forms_12.xml.

Defining the Validation Rules

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

Oracle CRM On Demand Desktop allows you to define which Oracle CRM On Demand fields are required to make sure the user completes the form correctly. When you define a field or set of fields as required, they are automatically highlighted in red.

> **Note:** Some fields in Microsoft Outlook are required by default, for example, the Full Name field for Contact records. When a user breaks the native Microsoft Outlook validation rule, the native Microsoft Outlook message is displayed and cannot be customized.

For more information, see *Correct Usage of the Forms_xx.XML File and Object ID*.

The following procedure describes how to define a validation rule so that both the Full Name and Account fields are required to save a Contact Record. If the user attempts to save a record without first entering or selecting values for these fields, then an error message is displayed.

In this procedure, you use the following files:

- **forms.js.** This file requires JavaScript.
- **package_res.xml.** This file requires an XML addition to specify the localized message that is displayed when a validation rule is violated.

To define validation rules

1. Open the forms.js file.
2. Locate the XML code that defines the fields and scripts used for the Contact form, for example, locate:

   ```
   'id="On Demand Contact"'
   ```

3. Define a validation object by adding the following code to the form script:

   ```
   var validation = ctx.validator;
   ```

   This object is used to define all the validation rules on a form.
4. Add the rule to this validation object by calling the validate_empty_field function on your validation object. This function accepts five arguments:

   ```
   validation.validate_empty_field(field, control, message_id, highlight_control,
   group);
   ```

   where:

   - **field.** This is the Oracle CRM On Demand field name you are going to specify as Required.
   - **control.** This is the ID of the control on a form that represents this field.

- **message_id.** This is the ID of a string resource in package_res.xml. The string with the specified ID is shown as an error message when the validation fails.
- **highlight_control.** If this is set to True, then the control will be highlighted if validation fails. The default value of this is True.
- **group.** The name of the validation group. It is not required.

5. Repeat Step 4 for each required field.
6. Open the package_res.xml file.
7. Identify where you want to add the new XML.

> **Tip:** Messages and labels are grouped together by comments. Validation rule messages are grouped under the `<!-- Script and validation Messages -->` comment.

8. Add a new <str> element to the XML code, for example:

```
<str key="msg_first_name_validation">First Name is required.</str>
```

The key value must match the message_id argument passed to the validate_unique_fields function.

Oracle CRM On Demand Desktop allows you to validate the format of Oracle CRM On Demand fields, for example, email address, and phone number, and so on. This feature prevents users from entering incorrect data into Oracle CRM On Demand records. If a user enters data in an incorrect format, then Oracle CRM On Demand Desktop displays an error message, and Oracle CRM On Demand Desktop does not save the data.

In this example, the Business Phone field is required to save a Contact record. If the user attempts to save a record with a value that is not in the correct format, then Oracle CRM On Demand Desktop displays an error message.

In this procedure, you use the following files:

- **forms.js.** This file requires JavaScript.
- **package_res.xml.** This file requires an XML addition to specify the localized message that is displayed when a validation rule is violated.

To define a validation rule for value formats

1. Open the forms.js file.
2. Locate the XML code that defines the fields and scripts used for the Contact form, for example:

```
'id="On Demand Contact"'
```

3. Define a validation object by adding the following code to the form script:

```
var validation = ctx.validator;
```

This object is used to define all the validation rules on a form.

4. Write a function that validates the phone number format.

This function must return True if the value in a field matches a specific pattern. Otherwise, it must return False. For example, this function uses regular expression to validate the Work Phone number (#) field format:

```
function validate_phone_number() {
 var phonetest = new RegExp(/^[+]?[0-9]{0,3}[\s]?[\(-. ]?[0-9]{2,3}[\)-.
]?[\s]?[0-9]{3}[-. ]?[0-9]{2}[-.]?[0-9]{2}$/);
 var snapshot = ctx.form.item.snapshot;
 return snapshot["Work Phone #"] != '' ? ( snapshot["Work Phone
#"].match(phonetest) != null ? true : false) : true
}
```

You can also use any other methods to validate the field format.

5. Add the rule to this validation object by calling the add_custom function on your validation object. This function accepts four arguments:

   ```
   validation.add_custom(custom_function, array_of_controls, message_id , group);
   ```

   where:

   - **custom_function.** This is the function you wrote in Step 4.
   - **array_of_controls.** This is the array of IDs of the control on a form that represents fields that must be highlighted if the custom rule is not followed.
   - **message_id.** This is the ID of a string resource in package_res.xml. The string with the specified ID is shown as an error message when a validation fails.
   - **group.** This is the name of the validation group. It is not required.

6. Open the package_res.xml file, and define the required messages.
7. Identify where you want to add the new XML.

   Tip: Messages and labels are grouped together by comments. Validation rule messages are grouped under the `<!-- Script and validation Messages -->` comment.

8. Add a new <str> element to the XML code, for example:

   ```
   <str key="msg_business_phone_validation">Business Phone format is not correct.</
   str>
   ```

   The key value must match the message_id argument passed into your custom function that you wrote in Step 4.

Adding Custom Logic

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

To improve usability, sometimes you must add custom logic to Oracle CRM On Demand Desktop. Oracle CRM On Demand Desktop allows you to manipulate Microsoft Outlook data, for example, you can do the following:

- Automatically include the ID of the current user in the Owner field.
- Cause the first letter of each word in the description of an activity to automatically capitalize.
- Change the state of some controls, for example, to inform the user that the control is required.

To add custom logic

1. Use an XML editor to open the forms_12.xml file.

   For more information, see *Files in the Customization Package*.

2. To add the custom logic for this example, add the following code:

   ```
   <form id="On Demand Activity"
   ```

```
 <script>
 <![CDATA[
 include("forms.js", "forms");
 var ctx = {
 "application": application,
 "ui": ui,
 "application_script": application_script,
 "form": form
 };
 var current_form = new
 forms.activity_form(forms.create_form_ctx(ctx));
 ]]>
 </script>
...
</form>
```

3. In forms.js, create the activity_form(ctx){} function.

   This function contains the custom logic.

Defining the Toolbar

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

For this example, it is recommended that you implement some actions for the custom object on a toolbar. An action can be simple, such as attaching a note to the custom object. An action can be more complex, such as sending an email to all contacts who are related to the custom object. In this example, you add the following buttons to the toolbar:

- Add Open in Oracle CRM On Demand
- Add attachment to the toolbar of the form
- Add New Activity to the Oracle CRM On Demand Desktop toolbar

To implement these changes, you modify the toolbars.xml file. Because this file typically already contains definitions for the Oracle CRM On Demand Desktop and form toolbars, you define only the custom buttons.

To define the toolbar

1. Encode the icon that represents the new button:
   a. Create a PNG file that includes the icon that will represent the button in the Oracle CRM On Demand Desktop client.
   b. Encode the PNG file for Base64.

      Because Oracle CRM On Demand Desktop stores the icon in the resource file in a Base64 encoded string, you must encode the PNG file. You can use any standard Base64 encoder, such as base64.exe.
   c. Remove the line breaks from the contents of the output file you created in Step b.
   d. Add the contents of the output file you created in Step b to the package_res.xml file.
2. To define a new button for the Oracle CRM On Demand Desktop toolbar, add the following code to the toolbars.xml file:

```
<toolbar caption="Oracle CRM On Demand Desktop" for="explorer">
  (…………….)
 <button name="New Activity" small_image="type_image:Activity:16">
 <action class="create_item" item_type="Action"/>
 </button>
  (…………….)
```

```
</toolbar>
```

3. To define a new button for the form toolbar, add the following code to the toolbars.xml file. Note that the code uses the term inspector to reference the form:

```
<toolbar caption="Oracle CRM On Demand Desktop" for="inspector">
 (…………….)
 <button name="Attach File" small_image="attach_btn_img">
 <action class="create_attachment" accept_type="Action">
 <attachment type="Attachment" name_field="Name" body_field="Body"
 linking_field="ParentId"/>
 </action>
 </button>
 (…………….)
</toolbar>
```

Defining Other Options

This task is a step in *Process of Customizing Objects in Oracle CRM On Demand Desktop*.

You can define other options.

To define other options

1. Define a support email address.

   For more information, see *Customizing the Email Address of the Support Team*.
2. Define the setup options.

   For more information, see *Third Party Options for Installing and Configuring Oracle CRM On Demand Desktop*.

Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop

This topic describes how to enable a custom object in Oracle CRM On Demand Desktop by modifying the metadata files in the customization package.

To enable custom object synchronization in Oracle CRM On Demand Desktop perform the following tasks:

1. *Defining the Custom Object*
2. *Defining Synchronization for the Custom Object*
3. *Defining a Mapping of the Custom Object*
4. *Defining a Mapping of the Custom Object if You Are Using DB FACADE Storage*
5. *Defining a Customized Microsoft Outlook User Interface for the Custom Object*
6. *Defining a Custom View for the Custom Object*
7. *Defining Labels Used as Resources for the Custom Object*
8. *Adding Custom Logic and Validation Rules in the Customized Microsoft Outlook Form*

Defining the Custom Object

This topic describes how to define the object name and field names for the object called Custom Object 1. The object name and field names are the same as those saved in the WDSL file for Custom Object 1.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To define the custom object

1. Using an XML editor, open the od_meta_info.xml file.
2. Locate the following <SiebelMetaInfo> XML element.:

```
<SiebelMetaInfo>

...

...

<object TypeId='Custom Object 1' Label='CustomObject1' LabelPlural='CustomObject1'
ViewMode='Sales Rep' IntObjName='Custom Object 1'
SiebMsgXmlElemName='CustomObject1'
SiebMsgXmlCollectionElemName='ListOfCustomObject1'>

</object>

</SiebelMetaInfo>
```

3. Update the following attributes in the <object> XML element with your custom object values:
   - **TypeId='Custom Object 1'.** The unique identifier of the object defined in the od_meta_info.xml file. The TypeId='Custom Object 1' attribute must have the same value as the IntObjName='Custom Object 1' attribute in the od_meta_info.xml file.
   - **IntObjName='Custom Object 1'.** The name of the integration object that Oracle CRM On Demand Desktop uses for requests. The IntObjName='Custom Object 1' attribute must have the same value as the TypeId='Custom Object 1' attribute.

     Note: For Custom Objects 1, 2, and 3, there is a space between the words in the IntObjName value, for example, Custom Object 1 or Custom Object 2, but for Custom Object 4 and custom objects with higher numbers, they are declared as one word, CustomObject4.

   - **SiebMsgXmlElemName='CustomObject1'.** The XML element of the object, which is declared in the "Custom Object 1" WSDL for Oracle CRM On Demand Web Services v2.0.
   - **SiebMsgXmlCollectionElemName='ListOfCustomObject1'.** The name of the collection XML element as declared in the "Custom Object WSDL for CRM OD Web Services v2.0.
4. Locate the following <object> element:

```
<object TypeId='Custom Object 1' Label='CustomObject1'
LabelPlural='CustomObject1' ViewMode='Sales Rep' IntObjName='Custom Object 1'
SiebMsgXmlElemName='CustomObject1'
SiebMsgXmlCollectionElemName='ListOfCustomObject1'>
 <field Name='CreatedBy' Label='CreatedBy' DataType='DTYPE_TEXT'
IsHidden="yes"/>
 <field Name='CreatedById' Label='CreatedById' DataType='DTYPE_ID'
IsRefObjId="yes" RefObjTypeId="User" IsHidden="yes"/>
<field Name='CreatedDate' Label='CreatedDate' DataType='DTYPE_UTCDATETIME'
IsHidden="yes"/>
```

```
</object>
```

5. Update the following Oracle CRM On Demand Desktop system fields as hidden fields for Microsoft Outlook:

   - **Name='CreatedById'.** The name of a field as declared in the "Custom Object 1" WSDL for CRM OD Web Services v2.0.
   - **IsHidden="yes".** This attribute indicates that the Oracle CRM On Demand Desktop system fields are hidden fields for Microsoft Outlook.
   - **'CreatedById' field.** The foreign key field that points to the User object in Oracle CRM On Demand Desktop.
   - **IsRefObjId="yes".** This <field> element attribute indicates that the CreatedById field is a foreign key field in Oracle CRM On Demand Desktop.
   - **RefObjTypeId="User".** This <field> element attribute indicates the name of the object type to which the CreatedById field points.
6. Locate the <object> element that you want to synchronize between Oracle CRM On Demand Desktop and Microsoft Outlook.
7. Update the following Custom Object 1 fields:

   ```
   <field Name='Id' Label='Id' DataType='DTYPE_ID' IsPrimaryKey='yes' />
   <field Name='Name' Label='Name' DataType='DTYPE_TEXT' />
   <field Name='Description' Label='Description' DataType='DTYPE_TEXT' />
   ```

   When the value for the IsPrimaryKey attribute is Yes, it indicates that this field is the primary key field for the object. For more information on the attributes defined in the <field> XML element, see *Field Element of the od_meta_info.xml File*.
8. Save and close the od_meta_info.xml file.

   The following code is an example of a complete object definition for Custom Object 1 in the od_meta_info.xml file:

   ```
   <SiebelMetaInfo>
   ...
   ...
    <object TypeId='Custom Object 1' Label='CustomObject1'
   LabelPlural='CustomObject1' ViewMode='Sales Rep' IntObjName='Custom Object 1'
   SiebMsgXmlElemName='CustomObject1'
   SiebMsgXmlCollectionElemName='ListOfCustomObject1'>
    <field Name='CreatedBy' Label='CreatedBy' DataType='DTYPE_TEXT'
    IsHidden="yes"/>
    <field Name='CreatedById' Label='CreatedById' DataType='DTYPE_ID'
    IsRefObjId="yes" RefObjTypeId="User" IsHidden="yes"/>
    <field Name='CreatedDate' Label='CreatedDate' DataType='DTYPE_UTCDATETIME'
    IsHidden="yes"/>
    <field Name='Id' Label='Id' DataType='DTYPE_ID' IsPrimaryKey='yes' />
    <field Name='Name' Label='Name' DataType='DTYPE_TEXT' />
    <field Name='Description' Label='Description' DataType='DTYPE_TEXT' />
    </object>
   </SiebelMetaInfo>
   ```

Related Topics

Specifying the Type of Object the User Can Confirm for Deletion

XML Code to Customize Synchronization

Defining Synchronization for the Custom Object

This topic describes how to define the synchronization of Custom Object 1 by editing the connector_configuration.xml file.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To define synchronization for the custom object

1. Using an XML editor, open the connector_configuration.xml file.
2. Locate the following <types> element:

   ```
   <types>
   ...
   ...
    <type id="Custom Object 1">
    ...
    </type>
   </types>
   ```

3. Define a new synchronization type in the <type> element.

   For more information on defining the synchronization type, see *Types Element of the connector_configuration.xml File*.
4. Define a <view> element inside the <type> element, for example:

   ```
   <type id="Custom Object 1">
    <view label="CustomObject1" label_plural="CustomObject1"
    small_icon="type_image:Generic:16"normal_icon="type_image:Generic:24"
    large_icon="type_image:Generic:48"></view>
   </type>
   ```

   For more information on defining a <view> element inside the <type> element, see *View Element of the connector_configuration.xml File*.
5. In the same code, define a <synchronizer> element with <link> tags, for example:

   ```
   <type id="Custom Object 1">
    <view label="CustomObject1" label_plural="CustomObject1"
    small_icon="type_image:Generic:16"normal_icon="type_image:Generic:24"
    large_icon="type_image:Generic:48"></view>
    <synchronizer name_format=":[:(Name) :]" threshold="10">
    <links></links>
    </synchronizer>
   </type>
   ```

   A <synchronizer> element is an element that describes additional, synchronization settings. It includes the <link> element, which describes references between types. In the previous example, you synchronize Custom Object 1 only as an independent parent object so that the values inside the <links> tags are blank. For more information on the <synchronizer> element, see *Synchronizer Element of the connector_configuration.xml File*.

   For more information on the <links> element, see *Links Element of the connector_configuration.xml File*.

   The following code is an example of a complete synchronization definition for Custom Object 1 in the connector_configuration.xml file:

   ```
   <type id="Custom Object 1">
   ```

```
 <view label="CustomObject1" label_plural="CustomObject1"
 small_icon="type_image:Generic:16"normal_icon="type_image:Generic:24"
 large_icon="type_image:Generic:48"></view>
 <synchronizer name_format=":[:(Name) :]" threshold="10">
 <links></links>
 </synchronizer>
</type>
```

6. Save and close the connector_configuration.xml file.

Related Topics

Specifying the Type of Object the User Can Confirm for Deletion

Defining Synchronization for a Custom Object

XML Code to Customize Synchronization

Defining a Mapping of the Custom Object

This topic describes how to define the mapping of Custom Object 1 in the od_basic_mapping.xml file.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To define a mapping of Custom Object 1

1. Using an XML editor, open the od_basic_mapping.xml file.
2. Locate the <types> element:

   ```
   <types>
   ...
   ...
    <type id="Custom Object 1" folder_type="10" display_name="CustomObject1"
    ver="2">
    </type>
   </types>
   ```

   In this example, id="Custom Object 1" is the unique identifier of the mapping definition for Custom Object 1 defined in the od_basic_mapping.xml file.
3. Define the new mapping type for Custom Object 1.

   For more information on the attributes used with the <type> element, see *Type Element of the od_basic_mapping.xml File*.
4. Define the <form> element inside the <type> element:

   ```
   <type id="Custom Object 1" folder_type="10" display_name="CustomObject1" ver="2">
    <form message_class="IPM.Contact.OnDemand.CustomObject1"
    display_name="CustomObject1" icon="type_image:Generic:16">OnDemand
    CustomObject1</form>
   </type>
   ```

   The values for this example are as follows:

   - **<form> element.** This value corresponds to the custom form ID value of the object's custom form, which is defined in the forms_xx.xml file.

- **"Custom Object 1"**. This value has a custom form definition with the value OnDemand CustomObject1. It is declared in the <form> element in od_basic_mapping.xml.

For information on the attributes for the form element, see *Form Element of the od_basic_mapping.xml File*.

5. In the following code, define the <custom_views> element:

```
<type id="Custom Object 1" folder_type="10" display_name="CustomObject1" ver="2">
 <form message_class="IPM.Contact.OnDemand.CustomObject1"
 display_name="CustomObject1" icon="type_image:Generic:16">OnDemand
 CustomObject1</form>
 <custom_views default_name = "#view_od_co1">
 <view id="all_customobject1" name ="#view_od_co1"> </view>
 </custom_views>
</type>
```

The <custom_view> element can contain many <view> elements. Similar to the value inside the <form> element in Step 4, the view id attribute value corresponds to the custom view value that is defined in the views.xml file. This example includes a view with the ID value of all_customobject1 which maps to a custom view.

> **Note:** The default_name attribute in the <custom_views> element and the name attribute in the <view> element has the following value: #view_od_co1. This label will be defined as a resource in the package_res.xml file.

For information on the attributes in the <custom_view> and <view> tags, see *Custom Views Element of the od_basic_mapping.xml File*. For information on declaring the custom view in the od_basic_mapping.xml file, see *Adding Custom Views in Microsoft Outlook*.

6. In the following code, map the Name field to a custom Microsoft Outlook field:

```
<type id="Custom Object 1" folder_type="10" display_name="CustomObject1" ver="2">
 ...
 ...
 <field id="Name">
 <reader>
 <mapi_user>
 <user_field id="od Name" ol_field_type="1"></user_field>
 <convertor>
 <string/>
 </convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="od Name" ol_field_type="1"></user_field>
 <convertor>
 <string/>
 </convertor>
 </outlook_user>
 </writer>
 </field>
</type>
```

This code maps the Name field you already defined in the od_meta_info.xml file to a custom Microsoft Outlook field. For information on the attributes in the <field> tags, see *Field Element of the od_basic_mapping.xml File*.

Each <field> element describes only one field. In the <field> element, there is a <reader> element and a <writer> element. For information on the <writer> element, see *Writer Element of the od_basic_mapping.xml File*.

7. In the following code, map the Description field to a native Microsoft Outlook field:

```
<type id="Custom Object 1" folder_type="10" display_name="CustomObject1" ver="2">
 ...
 ...
 <field id="Description">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A110000"></mapi_tag>
 <convertor>
 <string/>
 </convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="LastName"></outlook_field>
 <convertor>
 <string/>
 </convertor>
 </outlook_std>
 </writer>
 </field>
</type>
```

This code maps the Description field you already defined in the od_meta_info.xml file to the Microsoft Outlook field Last Name.

There are differences between this Description field mapping and the Name field mapping described in Step 6. The Name field is mapped to a custom Microsoft Outlook field, and it uses the <user_field> element in the <reader> and <writer> elements. However, because the Description field is mapped to a native Microsoft Outlook field, it uses the <outlook_field> element inside the <writer> element. For more information on Microsoft Outlook fields, see *Fields That Oracle CRM On Demand Desktop Uses for the Custom Object*. For sample code of a complete mapping definition for Custom Object 1 in the od_basic_mapping.xml file, see *XML Code That Maps a Definition for a Custom Object*.

8. Save and close the od_basic_mapping.xml file.

Related Topics

Specifying the Type of Object the User Can Confirm for Deletion

Defining the Custom Object

XML Code to Map a Field in Microsoft Outlook Storage

Defining a Mapping of the Custom Object if You Are Using DB FACADE Storage

If you are using DB FACADE storage to store the object, then you complete a simpler procedure instead of *Defining a Mapping of the Custom Object*.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To define a mapping of the custom object if you are using DB FACADE storage

1. Using an XML editor, open the od_basic_mapping.xml file.

2. Locate the <types> element:

```
<sd2_meta>
 <database>
 <types>
 ....
 <type id="Currency" icon="type_image:Currency:16">
 ...
 </type>
 </types>
 </database>
</sd2_meta>
```

3. Define the new mapping type for Custom Object 1.

```
<type id="Custom Object 1" icon="type_image:Custom Object 1:16"/>
```

4. Define the new field for the type. No mapping to a Microsoft Outlook field is required if an object is stored in DB FACADE.

```
<type id="Custom Object 1" icon="type_image:Custom Object 1:16"/>
 ...
 ...
 <field id ="Description">
 <type>
 <simple type="string"/>
 </type>
 </field>
 ...
</type>
```

5. Save and close the od_basic_mapping.xml file.

Related Topics

Specifying the Type of Object the User Can Confirm for Deletion

Defining the Custom Object

XML Code to Map a Field in Microsoft Outlook Storage

Defining a Customized Microsoft Outlook User Interface for the Custom Object

This topic describes how to define a customized Microsoft Outlook user interface for Custom Object 1 by editing the forms_12.xml file. The example in this topic is a basic customized form with two fields on the Microsoft Outlook user interface.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To define the form for Custom Object 1

1. Using an XML editor, open the forms_12.xml file.
2. Locate the following code:

```
<forms>
...
...
 <form id="OnDemand CustomObject1" >
 </form>
...
</forms>
```

3. In the <forms> tags, define the form.
4. In the following code, define the visible controls for the form:

```
<form id="OnDemand CustomObject1" >
 <page id="General" tag="0x10A6" min_height="155" min_width="520">
 <cell>
 <stack layout="vert" padding="5">
 <cell>
 <stack layout="horz" spacing="3">
 <!-- captions -->
 <cell size="65">
 <stack layout="vert" spacing="5">
 <cell size="21">
 <static id="lbl_co1_name" tab_order="2">
 <text>#lbl_co1_name</text>
 </static>
 </cell>
 <cell size="21">
 <static id="lbl_co1_description" tab_order="4">
 <text>#lbl_co1_description</text>
 </static>
 </cell>
 </stack>
 </cell>
 <!-- fields -->
 <cell>
 <stack layout="vert" spacing="5">
 <cell size="21">
 <edit id="Name" max_chars="100" tab_order="3">
 <field value="string">Name</field>
 </edit>
 </cell>
 <cell size="21">
 <edit id="Description" max_chars="100" tab_order="5">
 <field value="string">Description</field>
 </edit>
 </cell>
 </stack>
 </cell>
 </stack>
 </cell>
 </stack>
 </cell>
 </page>
</form>
```

For more information on customizing the forms_12.xml file, see *Defining the User Interface*.

Note: There is a section of cells for the captions, and a section of cells for the fields. You must structure the cells so that the captions align with the corresponding fields. In the captions section, there is reference to the lbl_co1_name and lbl_co1_description labels.

5. Edit the forms_12.xml file to hide the unused Microsoft Outlook controls.

This customization already contains new fields in the form, for example, the Name field. It also has repositioned Microsoft Outlook fields, for example, the LastName field is now repositioned to Description. The remaining Microsoft Outlook fields are not used in this example. However, they cannot be removed from the form definition. You can hide all the unused Microsoft Outlook controls by moving them to a location where the cell size value is zero. For more information on the Microsoft Outlook user interface, see *Defining the User Interface*. For an example of how the unused native Microsoft Outlook controls are removed in to a small cell, see *XML Code That Hides Unused Microsoft Outlook Controls*. For an example of a complete custom form for Custom Object 1 in the forms_12.xml file, see *XML Code of a Complete Form for Custom Object1*.

6. Save and close the forms_12.xml file.

Related Topics

Specifying the Type of Object the User Can Confirm for Deletion

Defining the User Interface

XML Code to Customize Forms

Defining a Custom View for the Custom Object

This topic describes how to define a custom view for Custom Object 1 by editing the views.xml file.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To define a custom view for the custom object

1. Using an XML editor, open the views.xml file.
2. Define a custom view for Custom Object 1.

   This custom view was already declared in the od_basic_mapping.xml file.
3. Create an <str> element in the views.xml file with the corresponding all_customobject1 key attribute.

   A custom view with the id="all_customobject1" attribute in the <view> element is declared in the first line of the following code:

```
<str key="all_customobject1">
<![CDATA[<?xml version="1.0"?>
 <view>
 <viewname>$view_od_co1$</viewname>
 <viewstyle>table-layout:fixed;width:100%;font-family:Segoe UI;fontstyle:
 normal;font-weight:normal;font-size:8pt;color:Black;font-charset:0</
 viewstyle>
 <viewtime>0</viewtime>
 <linecolor>8421504</linecolor>
 <linestyle>3</linestyle>
 <gridlines>1</gridlines>
 <collapsestate/>
 <rowstyle>background-color:window;color:windowtext</rowstyle>
 <headerstyle>background-color:#D3D3D3</headerstyle>
 <previewstyle/>
 <arrangement>
 <autogroup>0</autogroup>
 <collapseclient/>
 </arrangement>
 <multiline>
 <width>0</width>
 </multiline>
 <column>
 <type>string</type>
 <heading>Name</heading>
 <prop>http://schemas.microsoft.com/mapi/string/{00020329-0000-0000-C000-
 000000000046}/od%20Name</prop>
 <width>50</width>
 <style>padding-left:3px;;text-align:left</style>
 <editable>1</editable>
 <userheading>Name</userheading>
 </column>
 <column>
 <type>string</type>
 <heading>Description</heading>
 <prop>urn:schemas:contacts:sn</prop>
 <width>426</width>
 <style>padding-left:3px;;text-align:left</style>
 <editable>1</editable>
 <userheading>Description</userheading>
 </column>
 <orderby>
 <order>
 <heading>Name</heading>
 <prop>http://schemas.microsoft.com/mapi/string/{00020329-0000-0000-C000-
 000000000046}/od%20Name</prop>
 <type>string</type>
 <sort>asc</sort>
 </order>
 </orderby>
 <groupbydefault>0</groupbydefault>
 <previewpane>
 <markasread>0</markasread>
 </previewpane>
 </view>]]>
</str>
```

The views.xml file defines columns for displaying objects in grids. In this example, the all_customobject1 view displays all custom objects in the top-level folder in Microsoft Outlook.

In the previous example code, you see a column element in the custom view for the Name field and another column element for the Description field. Within the column element, there is a prop element. The prop element provides the object property name, and is written in a special format. All Microsoft Outlook custom fields, like the Name field, are in the following format:

http://schemas.microsoft.com/mapi/string/{00020329-0000-0000-C000-000000000046}/"

The Name field is then appended to the Microsoft Outlook custom field the with the user_field id from the od_basic_mapping.xml.

Note: Any spaces in the text string must be replaced with %20.

All Microsoft Outlook native fields take the following format:

```
urn:schemas:contacts:sn" (LastName), "urn:schemas:contacts:givenName" (FirstName)
```

For more information on native Microsoft Outlook properties, see your Microsoft documentation.

4. Save and close the views.xml file.

Related Topics

Specifying the Type of Object the User Can Confirm for Deletion

Adding Custom Views in Microsoft Outlook

XML Code to Customize Views Used for Microsoft Outlook Explorer List

Defining Labels Used as Resources for the Custom Object

This topic describes how to define labels used as resources for the custom object called Custom Object 1 in the package_res.xml file.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To define the labels used as resources for Custom Object 1

1. Using an XML editor, open the package_res.xml file.
2. Define a custom view for Custom Object 1.
3. Define the labels used as resources for Custom Object 1 in the package_res.xml file.

   These labels are used in the od_basic_mapping.xml file and in the forms_12.xml file.
4. Include the following labels in the package_res.xml file:

```
<res_root>
...
...
 <!-- C01 form -->
 <str key="lbl_co1_name">Name</str>
 <str key="lbl_co1_description">Description</str>
 <str key="view_od_co1">CustomObject1</str>
...
...
</res_root>
```

5. Save and close the package_res.xml file.

Related Topics

Specifying the Type of Object the User Can Confirm for Deletion

Customizing the Email Address of the Support Team

Adding Custom Logic and Validation Rules in the Customized Microsoft Outlook Form

This topic describes how to add custom logic and validation rules for certain fields in the customized Microsoft Outlook form for Custom Object 1.

This task is a step in *Process of Enabling Custom Object Synchronization in Oracle CRM On Demand Desktop*.

To add custom logic and validation rules

1. Using an XML editor, open the forms_12.xml file.
2. Edit the <script> element as follows:

```
<form id="OnDemand CustomObject1" >
 <script>
 <![CDATA[
 include("forms.js", "forms");
 var ctx = {
 "application": application,
 "ui": ui,
 "application_script": application_script,
 "form": form
 };

 var current_form = new forms.od_co1_form(forms.create_form_ctx(ctx));

 ]]>
 </script>
 <page id="General" tag="0x10A6" min_height="155" min_width="520">
 ..
 ..
 </page>
</form>
```

Note: A JavaScript function called od_co1_form() is in the <script> element.

3. Save and close the forms_12.xml file.
4. Using a JavaScript editor, open the forms. js file.
5. Create the od_co1_form() function, using the following example:

```
function od_co1_form(ctx)

{
 //FORM VALIDATION
 var validator = ctx.validator;
 validator.validate_empty_field("Name", "Name", "msg_co1_name_validation", false);
}
```

6. Define a new label, msg_co1_name_validation in the od_co1_form() function.

   For more information, see *Defining the Validation Rules*.
7. Add the msg_co1_name_validation label to the package_res.xml file:

```
<res_root>
..
..
 <!-- C01 form -->
 <str key="lbl_co1_name">Name</str>
 <str key="lbl_co1_description">Description</str>
 <str key="view_od_co1">CustomObject1</str>
 <str key="msg_co1_name_validation">C01 Name is required.</str>..
..
</res_root>
```

8. Save and close the forms.js file.

Related Topics

XML Files in the Customization Package

Defining the Validation Rules

Adding Custom Logic

Process of Adding a Predefined Oracle CRM On Demand Picklist to Oracle CRM On Demand Desktop

This topic gives one example of how to add a predefined Oracle CRM On Demand picklist to Oracle CRM On Demand Desktop. You might use this feature differently, depending on your business model. In this example, you add a picklist to the Contact form, which allows the user to choose from a set of predefined contact methods. These methods indicate how the contact prefers to be contacted, such as through email, pager, or phone.

To add a predefined Oracle CRM On Demand picklist to Oracle CRM On Demand Desktop, do the following:

1. *Adding a Field to the Customization Package*
2. *Customizing the Physical Layout for the Picklist*
3. *Publishing and Testing a Custom MVG Field*

Adding a Field to the Customization Package

This task is a step in *Process of Adding a Predefined Oracle CRM On Demand Picklist to Oracle CRM On Demand Desktop*.

In this topic you add a field to the customization package.

To add a field to the customization package

1. Create a working set of files for the customization package:
   a. Open a DOS command prompt, and then navigate to the directory that contains the current files of the customization package.

      For more information, see *Files in the Customization Package*.
   b. Create a copy of the current set of customization package files.
   c. Move the original set of files to a backup directory.

      If it is necessary to restore the configuration that existed before you started this customization effort, then you can revert to this backup set of files.
   d. To create a working set of customization package files, rename the set of files you copied in Step b, for example, enter the following command:

      ```
      rename v01* v02*
      ```

      This command renames the prefix for all files in the directory that currently use v01 as the prefix. For example, it renames v01_forms_12.xml to v02_forms_12.xml. It is recommended that you use this technique to indicate that you have modified the customization package.
2. To define the picklist, add the following element to the od_meta_info.xml file:

```
<picklist TypeId='PickList_Preferred_Communications' CollectionTypeFld
Name='Type' SrcObjectTypeId='PickList_Preferred_Communications'
ValueFldName='Value' LabelFldName='Value' LangFldName='Language' >
 <master_filter_expr>
 <![CDATA[
 [Parent] = LookupValue ('OFFER_MEDIA', 'Package')
```

```
  ]]>
  </master_filter_expr>
</picklist>
```

For more information, see *Defining Attributes of the Picklist Element*.

3. Add the Preferred Communications field to the Contact object:

   a. To locate the Contact object, search for the following text:

   ```
   object TypeID='Contact'
   ```

   b. Add the following element to the Contact object:

   ```
   <field Name="Preferred Communications" Label="Preferred Communications"
   DataType="DTYPE_TEXT" HasPicklist="yes" PicklistIsStatic="yes"
   PicklistCollectionType="OFFER_MEDIA" PicklistTypeId="PickList Preferred
   Communications" IOElemName="PreferredCommunications" />
   ```

4. Repeat Step 2 for each of the following objects:

   - Account.Contact
   - Opportunity.Contact

   In this example, these objects in the od_meta_info.xml file must include the Preferred Communications field. You must add this field to each object.

5. Add code that creates a map for the picklist between the Oracle CRM On Demand server and Oracle CRM On Demand Desktop for the parent Contact object:

   a. Use an XML editor to open the od_basic_mapping.xml file.

   b. Create a new object type for the picklist.

   For more information, see *Code to Create a New Object Type for the Picklist*.

   c. To locate the parent object, search the od_basic_mapping.xml file for the following code:

   ```
   <type id="Contact"
   ```

   d. Add code to the Contact object that defines a map between the Oracle CRM On Demand server and Oracle CRM On Demand Desktop.

   For more information, see *Code to Define a Mapping Between the Oracle CRM On Demand Server and Oracle CRM On Demand Desktop*.

6. Add code that creates a map for the picklist between the Oracle CRM On Demand server and Oracle CRM On Demand Desktop for the child Account Contacts object:

   a. Choose the code from the contact object that you use to map the child account object.

   For more information, see *Mapping Child Objects for a Custom Picklist*.

   b. Copy this code to the clipboard.

   c. To locate the Account Contacts child object, search the od_basic_mapping.xml file for the following code:

   ```
   type id="Account.Contact.Association"
   ```

   d. To locate the Contact field, search in the Account.Contact.Association object for the following text:

   ```
   field id="ContactId"
   ```

e. To locate the matching ContactId, search the ContactId field for the following text:

```
user_field id="OD Contact ID"
```

f. Paste the contact fields that you copied to the clipboard in Step a into the OD Contact ID user field.

For more information, see *Mapping Child Objects for a Custom Picklist*.

g. To map the Preferred Communications field, add the following code immediately after the code you pasted in Step f:

```
<field from="Preferred Communications" to="ContactPreferred
Communications"></field>
```

7. Repeat Step 5 for the opportunity child object.

Defining Attributes of the Picklist Element

If you define a picklist element in the od_meta_info.xml file, then the TypeId and SrcObjectTypeId attributes of this element must match the value in the PicklistTypeId property of the object definition. For example, assume you add the following field:

```
<field Name='Note Type' Label='#fld_account_account_note@note_type'
DataType='DTYPE_TEXT' HasPicklist='yes' PicklistIsStatic='yes'
PicklistTypeId='AccountNoteType' IOElemName='NoteType' />
```

In this example, you must set the TypeId attribute in the picklist element to AccountNoteType.

The following table describes the values to use in the picklist element.

Adjust these values so they match the field names on the integration component field.

Attribute	Value
CollectionTypeFldName	Type
ValueFldName	Value
LangFldName	Language

To add more filters, you can use the master_filter_expr attribute. In the example on Step 2 in the previous topic, the master_filter_expr attribute constrains the values to the correct LOV Type.

Code to Create a New Object Type for the Picklist

To create a new object type for the picklist, you must use the following format for the type id attribute:

```
<type id="object_name
 field_namePicklist" predefined_folder="1">
```

where:

- object_name is the name of the object type in the od_meta_info.xml file.
- field_name is the name of the field that resides in the object that you define in object_name.

For example:

```
<type id="ContactPreferred CommunicationsPicklist" predefined_folder="1">
```

To create a new object type for a picklist, add the following code to the od_basic_mapping.xml file:

```
 <type id="ContactPreferred CommunicationsPicklist" predefined_folder="1">
 <form message_class="IPM.Contact.OnDemand.ContactPreferred
CommunicationsPicklist"></form>
 <field id="Label">
 <reader>
 <mapi_user>
 <user_field id="On Demand picklistLabel" ol_field_type="1"></ user_field>
 convertor><string/></convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="On Demand picklistLabel" ol_field_type="1"></user_field>
 <convertor><string/></convertor>
 </outlook_user>
 </writer>
 </field>
 <field id="Value">
 <reader>
 <mapi_user>
 <user_field id="On Demand picklistValue" ol_field_type="1"></user_field>
 <convertor><string/></convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="On Demand picklistValue" ol_field_type="1"></user_field>
 <convertor><string/></convertor>
 </outlook_user>
 </writer>
 </field>
 <field id="SortOrder">
 <reader>
 <mapi_user>
 <user_field id="od SortOrder" ol_field_type="3"></user_field>
 <convertor><integer/></convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="od SortOrder" ol_field_type="3"></user_field>
 <convertor><integer/></convertor>
 </outlook_user>
 </writer>
 </field>
 <field id="IsDefault">
 <reader>
 <mapi_user>
 <user_field id="od IsDefault" ol_field_type="6"></user_field>
 <convertor><bool/></convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="od IsDefault" ol_field_type="6"></user_field>
 <convertor><bool/></convertor>
 </outlook_user>
 </writer>
 </field>
</type>
```

Code to Define a Mapping Between the Oracle CRM On Demand Server and Oracle CRM On Demand Desktop

To define a map between the Oracle CRM On Demand server and Oracle CRM On Demand Desktop, add the following field element to the Contact object of the od_basic_mapping.xml file:

```
<field id="Preferred Communications">
 <reader> <mapi_user>
 <user_field id="od Preferred Communications" ol_field_type="1"></user_field>
 <convertor><string/></convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="od Preferred Communications" ol_field_type="1"></user_field>
 <convertor><string/></convertor>
 </outlook_user>
 </writer>
</field>
```

If you are adding your object to Oracle CRM On Demand Desktop's DB FACADE storage, then add this code:

```
<field id = "Preferred Communications">
 <type>
 <simple type="string"/>
 </type>
</field>
```

If you are adding a picklist object to DB FACADE storage, which is recommended for picklists, then you do not have to specify the `predefined_folder` attribute. Generally, the XML code to add picklists is as follows:

```
<type id="ContactPreferred CommunicationsPicklist"
icon="type_image:ContactPreferredCommunicationsPicklist:16">
 <field id ="Label">
 <type>
 <simple type="string"/>
 </type>
 </field>
 <field id ="Value">
 <type>
 <simple type="string"/>
 </type>
 </field>
 <field id ="SortOrder">
 <type>
 <simple type="integer"/>
 </type>
 </field>
 <field id ="IsDefault">
 <type>
 <simple type="boolean"/>
 </type>
 </field>
</type>
```

Mapping Child Objects for a Custom Picklist

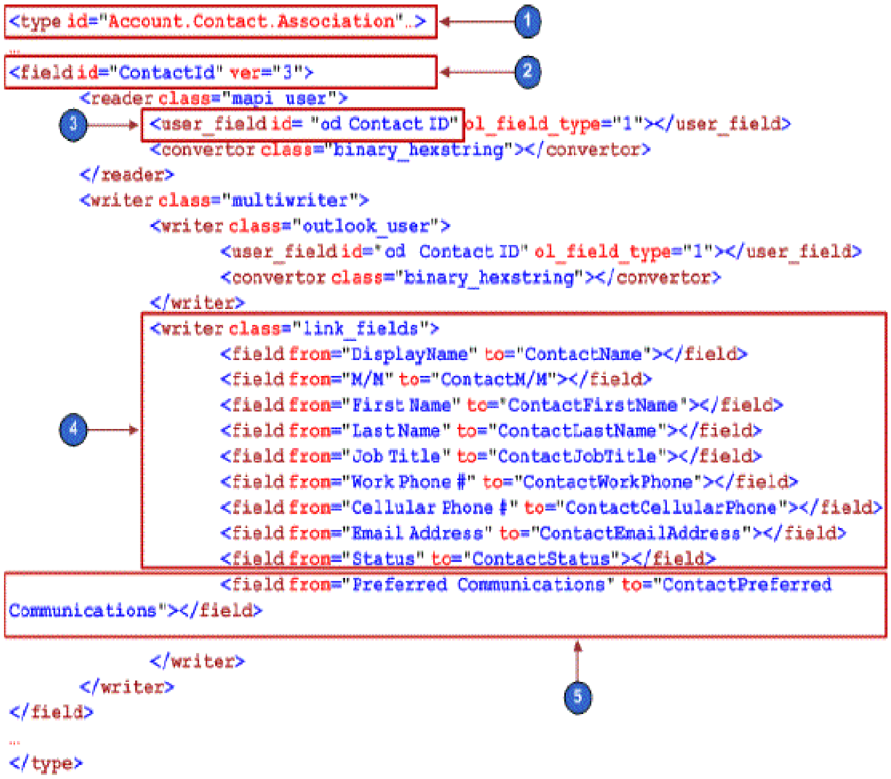
It is recommended that you do not map a child object directly in the child object. Instead, copy values from the parent object, and then paste them into the child object. This technique provides the following advantages:

- Allows Oracle CRM On Demand Desktop to copy values on the contact to the child object, such as the account or opportunity

- If the user changes the value in a contact, then Oracle CRM On Demand Desktop automatically updates the child objects

Example Code to Map Child Objects for a Custom Picklist

The following figure illustrates example code to map a child object for a custom picklist.



```
<type id="Account.Contact.Association"..>
...
<field id="ContactId" ver="3">
     <reader class="mapi user">
          <user_field id= "od Contact ID" ol_field_type="1"></user_field>
          <convertor class="binary_hexstring"></convertor>
     </reader>
     <writer class="multiwriter">
          <writer class="outlook_user">
               <user_field id="od  Contact ID" ol_field_type="1"></user_field>
               <convertor class="binary_hexstring"></convertor>
          </writer>
          <writer class="link_fields">
               <field from="DisplayName" to="ContactName"></field>
               <field from="M/M" to="ContactM/M"></field>
               <field from="First Name" to="ContactFirstName"></field>
               <field from="Last Name" to="ContactLastName"></field>
               <field from="Job Title" to="ContactJobTitle"></field>
               <field from="Work Phone #" to="ContactWorkPhone"></field>
               <field from="Cellular Phone #" to="ContactCellularPhone"></field>
               <field from="Email Address" to="ContactEmailAddress"></field>
               <field from="Status" to="ContactStatus"></field>
               <field from="Preferred Communications" to="ContactPreferred
Communications"></field>

          </writer>
     </writer>
</field>
...
</type>
```

Explanation of Callouts

The example code to map a child object for a custom picklist includes the following items:

1. The following attribute identifies the account child object of the parent contact:

   ```
   type id="Account.Contact.Association"
   ```

2. The following element identifies the Contact field in the account object:

   ```
   field id="ContactId"
   ```

3. The following attribute identifies the matching ContactId:

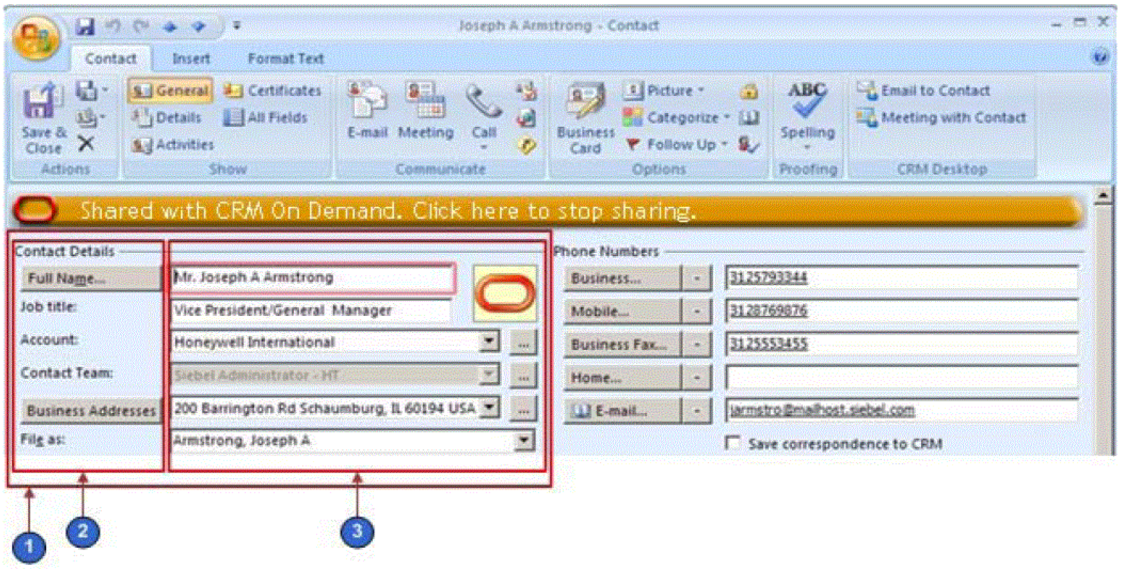
   ```
   user_field id="od Contact ID"
   ```

4. You copy these fields from the parent contact object and then paste them into the account child object.
5. You add the Preferred Communications field to provide a way to add it to Account Contact forms. You define this field in *Code to Define a Mapping Between the Oracle CRM On Demand Server and Oracle CRM On Demand Desktop*.

Customizing the Physical Layout for the Picklist

This task is a step in *Process of Adding a Predefined Oracle CRM On Demand Picklist to Oracle CRM On Demand Desktop*.

The following figure illustrates the Contact Details section of the contact form that you customize in this example.



Shared with CRM On Demand. Click here to stop sharing.

Contact Details

Full Name... Mr. Joseph A Armstrong

Job title: Vice President/General Manager

Account: Honeywell International

Contact Team: Siebel Administrator - HT

Business Addresses 200 Barrington Rd Schaumburg, IL 60194 USA

File as: Armstrong, Joseph A

Phone Numbers

Business... 3125793344

Mobile... 3128769876

Business Fax... 3125553455

Home...

E-mail... jarmstro@mailhost.siebel.com

□ Save correspondence to CRM

Explanation of Callouts

The Contact form includes the following parts:

1. The following object in the forms_12.xml file identifies the Contact Details section:

   ```
   OnDemand Contact
   ```

   Oracle CRM On Demand Desktop considers this area as a single cell. This cell includes two child regions that are placed horizontally in relationship to one another.
2. The left side of this cell includes six cells that are arranged vertically.
3. The right side of the cell includes the following items:

   - The Oracle oval link and the ellipsis (. . .) buttons include their own cell layers.
   - The top cell includes the following items:
     - Two horizontal subcells. These subcells accommodate the Oracle oval link.
     - Two vertical subcells that accommodate the fields to the left of the Oracle oval link.
   - The remaining cells below the top cell include subcells that accommodate their picklists and ellipsis (. . .) buttons.

To customize the physical layout for the picklist

1. Examine the physical location of where to place the custom field:

   a. Open the Oracle CRM On Demand Desktop client, and then navigate to the Contact form.

For more information, see the figure in the previous topic.

b. In the Contact Details section, locate the Business Addresses field.

   In this example, you add the custom Preferred Contact Method field under the existing Business Address field.

2. To provide sufficient vertical room for the custom field and label, increase the cell size:

   a. Locate the cell size attribute for the cell that you want to increase.

      You must increase the size of the cell that contains all the other cells. In this example, examine the figure in the previous topic, and then note that you must increase the height of the cells that are labeled 2 and 3. To do this, you increase the size of the cell that is labeled 1.

   b. Increase the cell size by 30.

      For example, if the current cell size is 155, then change it to 185:

```
<cell size="185">
```

3. Add the label for the custom field:

   a. Use an XML editor to open the forms_12.xml file.

      In this example, assume you are using Microsoft Outlook 2007 or above. For more information, see *Customizing a Form*.

   b. To locate the form you must modify and locate the following code:

```
form id="OnDemand Contact"
```

      Each form includes the OnDemand prefix, which is immediately followed by the object name, such as Contact.

   c. Locate the following code, which defines the label for the existing business address:

```
<cell size="22">
 <control id="dd_addresses" class="dropdown"
 caption="#head_business_addresses" tab_order="12" />
</cell>
```

   d. The code for the label and fields is located in the OnDemand Contact object after the validation rules.

      To define the label for the new contact method, add the following code immediately below the code you located in Step c:

```
<cell size="22">
 <static id="ContactMethod" tab_order="9">
 <text>#lbl_ContactMethod</text>
 <static>
</cell>
```

      For more information, see *Format of the Label for a Custom Field*.

4. Add the custom field:

   a. Locate the following code, which defines the field for the existing business address. For brevity, ellipsis points (...) indicate code that has been omitted:

```
<cell size="21">
 <stack layout="horz" spacing="5">
 <cell>
 <combobox id="business_address_mvg" tab_order="13">
 <field>Primary Address Id</field>
 . . .
 </combobox>
 </cell>
 </stack>
</cell>
```

To simplify this step, search for unique text, such as business_address_mvg.

b. To define the field for the new contact method, add the following code immediately below the code you located in Step a:

```
<cell size="22">
 <combobox id="cbx_ContactPreferred CommunicationsPicklist">
 <field>Preferred Communications</field>
 <source type="ContactPreferred CommunicationsPicklist" field="Value"
format=":[:(Label):]"> </source>
 </combobox>
</cell>
```

For more information, see *Format of the Custom Field*.

5. Define the custom symbolic strings:

a. Use an XML editor to open the package_res.xml file.

b. Add the following code anywhere in the file:

```
<str key="lbl_ContactMethod">Contact Method:</str>
<str key="head_contact_method">Contact Method</str>
```

You can place these strings anywhere in the file. To assist with maintenance, it is recommended that you place them with similar strings. If it is necessary, then you can create alternate package_resource_xml files to create symbolic strings that accommodate another language. For more information, see *Customizing Localization*.

Format of the Label for a Custom Field

To define the label for the custom contact method field, you add the following code:

```
<cell size="22">
 <static id="ContactMethod" tab_order="9">
 <text>#lbl_ContactMethod</text>
 </static>
</cell>
```

where:

- Id is an arbitrary, unique value for the form
- Tab_order determines the order in which Oracle CRM On Demand Desktop places the cursor in the fields in the form when the user presses the TAB key
- Text defines the text that Oracle CRM On Demand Desktop uses for the label
- # indicates the symbolic string that is defined in the package_res.xml file. You can use this string for a global deployment.

Format of the Custom Field

To define the custom field for the contact method, you add the following code:

```
<cell size="22">
```

```
     <combobox id="cbx_ContactPreferred CommunicationsPicklist">
     <field>Preferred Communications</field>
     <source type="ContactPreferred CommunicationsPicklist" field="Value"
     format=":[:(Label):]"> </source>
     </combobox>
    </cell>
```

where:

- Id is an arbitrary, unique value in the form. A picklist must include the cbx prefix.
- Tab_order determines the order in which Oracle CRM On Demand Desktop places the cursor in the fields in the form when the user presses the TAB key. The value that you enter must be greater than the value you enter in the Tab_order for the label.
- Source determines where to get the data
- Type identifies the object name. This name is defined in the od_basic_mapping.xml file.
- Field specifies the field that provides the values that the user chooses from the picklist. In this example, the Value field provides these values.
- <Field> identifies the field name from the object definition that populates the picklist.
- Format specifies how to display text in the picklist.

The format element allows you to use a combination of static text and fields in the picklist. It uses the following format:

```
*any_static_text*:[:(field1_name):]*any_static_text*:[:(field2_name):]*any_static_
text*
```

You must use the brackets ([ ]), colon (:), and parentheses ( ), or example:

```
Contact :[:(First Name):] :[:(Last Name):] ? Contact: John Smith
```

Publishing and Testing a Custom Picklist

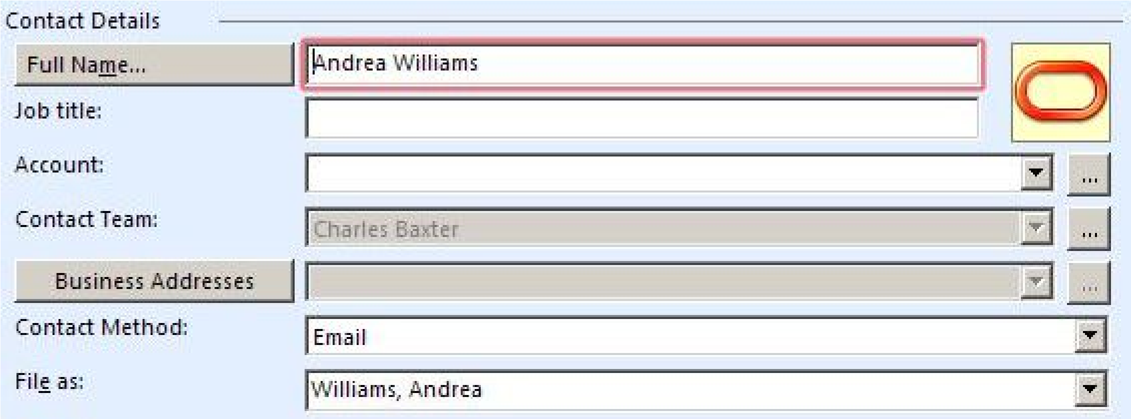
This task is a step in *Process of Adding a Predefined Oracle CRM On Demand Picklist to Oracle CRM On Demand Desktop*.

In this topic, you publish and test your customization.

To publish and test a custom picklist

1. Publish your changes.

   For more information, see *Publishing or Unpublishing Customization Packages*.
2. Test your changes:
   a. Open the Oracle CRM On Demand Desktop Client, and then navigate to the Contact form.
   b. Verify that the form includes a label and picklist for the Contact Method field, as illustrated in the following diagram:



Contact Details

Full Name...	Andrea Williams
Job title:	
Account:	
Contact Team:	Charles Baxter
Business Addresses	
Contact Method:	Email
File as:	Williams, Andrea

c. Click the down arrow next to the Contact Method field, and then verify that Oracle CRM On Demand Desktop displays a picklist that contains the following values:

- None
- Chat
- Email
- Fax
- Pager
- Phone
- Wireless message

d. Choose a value in the picklist, and then verify that Oracle CRM On Demand Desktop changes the value in the Contact Method field to the value you choose.

Roadmap for Adding an MVG Field

This topic gives one example of how to add an MVG field. You might use this feature differently, depending on your business model.

To add an MVG field, do the following task and process:

1. *Process of Modifying the Customization Package to Add an MVG*
2. *Publishing and Testing a Custom MVG Field*

An MVG field displays an association with other objects. This association can be a many-to-many association or a many-to-one association. The user can use an MVG field to do the following:

- Add or remove an association
- Change a primary association
- Browse existing associations

You can also use an MVG to reduce the layout of a form. You can display a set of associated records in a single-line control instead of using the outlook_view control. In this example, you add an MVG control to the Opportunity form. This MVG displays an association between an opportunity and channel partners.

Process of Modifying the Customization Package to Add an MVG

This task is a step in *Roadmap for Adding an MVG Field*.

To modify the customization package to add an MVG, do the following:

1. *Adding a Custom Object*
2. *Adding the MVG Link*
3. *Adding the Primary Field*
4. *Adding a Field*
5. *Adding a Lookup View*
6. *Code to Define a Sales Book Control*
7. *Adding a Label, Button, and Selector Control*
8. *Customizing the Validation Message and Labels*

Adding a Custom Object

This task is a step in *Process of Modifying the Customization Package to Add an MVG*.

To add a custom object to Oracle CRM On Demand Desktop, you display the object in Oracle CRM On Demand and then modify customization package XML files. In this example, you make available and then add the Channel Partner object.

To add a custom object

1. Make sure the object you must add is available.
2. To add a custom object type, modify the od_meta_info.xml file.
   For more information, see *Code to Add a Custom Object Type*.
3. To map objects, modify the od_basic_mapping.xml file.
   For more information, see *Code to Map a Custom Object*.
4. To configure synchronization for the custom object, modify the connector_configuration.xml file.
   For more information, see *Code to Configure the Synchronization of a Custom Object*.

Adding the MVG Link

This task is a step in *Process of Modifying the Customization Package to Add an MVG*.

In this topic, you add the MVG link to the business logic file.

To add the MVG link

1. Use a JavaScript editor to open the business_logic.js file.
   For more information, see *Files in the Customization Package*.
2. Locate the following function:
   create_ondemand_meta_scheme2
3. Add the code to add a new association.
   For more information, see *Code to Add a New Association*.

Adding the Primary Field

This task is a step in *Process of Modifying the Customization Package to Add an MVG*.

In this topic, you add the primary field to the customization package. This field displays in the MVG dialog box that you add for this example.

To add the primary field

1. To add the primary field, do the following:
   a. Use an XML editor to open the od_meta_info.xml file.
   b. Add the following code to the Opportunity object in the od_meta_info.xml file:

   ```
   <field Name='Primary Partner Id' Label='Primary Partner Id'
   DataType='DTYPE_ID' IsFilterable='no' IsRefObjId='yes'
   RefObjTypeId='ChannelPartner' IOElemName='PrimaryPartnerId' />
   ```

   This code adds the field, Primary Partner Id, to the Opportunity type. The field is defined as the reference for ChannelPartner type.
   c. Save and then close the od_meta_info.xml file.
   d. Use an XML editor to open the od_basic_mapping.xml file:
   e. Add the following code to the Opportunity type element:

   ```
   <field id="Primary Partner Id">
    <reader>
    <mapi_user>
    <user_field id="od Primary Partner Id" ol_field_type="1"></user_field>
    <convertor><binary_hexstring/></convertor>
    </mapi_user>
    </reader>
    <writer>
    <outlook_user>
    <user_field id="od Primary Partner Id" ol_field_type="1"></user_field>
    <convertor><binary_hexstring/></convertor>
    </outlook_user>
    </writer>
   </field>
   ```

   This code adds custom field Primary Partner Id to Opportunity type to access the field from Microsoft Outlook. The field is defined as binary_hexstring.
2. Add the link:
   a. Use an XML editor to open the connector_configuration.xml file.
   b. Add the following code to the Opportunity type element:

   ```
   <link>Primary Partner Id</link>
   ```

   This code is required for defining foreign key fields; that is, for defining links to other objects.
   c. Save and then close the connector_configuration.xml file.

Adding a Field

This task is a step in *Process of Modifying the Customization Package to Add an MVG*.

This topic describes how to add the ChannelPartnerStatus field.

To add a field

1. Use an XML editor to open the od_basic_mapping.xml file.

2. Add the following code to the Opportunity.Channel_Partner.Association type in the od_basic_mapping.xml file.

   ```
   <field id="ChannelPartnerStatus">
    <reader>
    <mapi_user>
    <user_field id="od ChannelPartnerStatus" ol_field_type="1"></user_field>
    <convertor><string/></convertor>
    </mapi_user>
    </reader>
    <writer>
    <outlook_user>
    <user_field id="od ChannelPartnerStatus" ol_field_type="1"></user_field>
    <convertor><string/></convertor>
    </outlook_user>
    </writer>
   </field>
   ```

3. Save and then close the od_basic_mapping.xml file.

Adding a Lookup View

This task is a step in *Process of Modifying the Customization Package to Add an MVG*.

To customize a SalesBook control, you use the definition of a lookup view. A SalesBook control displays a list of objects and then allows the user to choose any object from this list. Oracle CRM On Demand Desktop uses this control on forms and MVG dialog boxes. In this example, you add the definition for a lookup view for Channel Partner objects.

To add a lookup view

1. Use an XML editor to open the lookup_view_defs.xml file.
2. Add the following code to the array element that contains the lookup types in the lookup_view_defs.xml file:

   ```
   <item value="ChannelPartner"></item>
   ```

3. Add the definition for the lookup view.

   For more information, see *Code to Add a Lookup View*.

Adding a Label, Button, and Selector Control

This task is a step in *Process of Modifying the Customization Package to Add an MVG*.

In this topic, you add a label, button, and selector control.

To add a label, button, and selector control

1. Use an XML editor to open the forms_xx.xml file.
2. Add a label for the MVG:
3. Locate the following section:

   ```
   right side captions
   ```

4. Insert the following code immediately under the code that defines the Probability label:

   ```
   <cell size="22">
    <static id="0x20014" tab_order="166">
    <text>#lbl_channel_partner</text>
    </static>
   </cell>
   ```

5. Add the button and primary selector control:

a. Locate the following section:

```
right side fields
```

b. Add the following code immediately under the code that defines the Probability control.

```
<cell size="22">
 <stack layout="horz">
 <cell>
 <mvg_primary_selector id="channel_partner_mvg">
 <source type="Opportunity.Channel_Partner.Association"
 left_id="OpportunityId" item_value="ChannelPartnerId"
 display_format=":[:(PartnerName):]"></source>
 <field>Primary Partner Id</field>
 </mvg_primary_selector>
 </cell>
 <cell size="5"></cell>
 <cell size="22">
 <button id="btn_mvgChannelPartner">
 <text>...</text>
 </button>
 </cell>
 </stack>
</cell>
```

Make sure you specify the Id for the button and the Id for the primary selector in the same way that you specified them in the script.

6. Increase the size of the cell that contains the label, button, and selector control:

a. Locate the following section:

```
Category bar
```

b. Locate the fifth cell that is included in the Category bar section.

c. Change the code of the fifth cell to the following code:

```
<cell size="153">
```

7. Save and then close the forms_xx.xml file.

Customizing the Validation Message and Labels

This task is a step in *Process of Modifying the Customization Package to Add an MVG*.

In this topic, you customize the validation message and a label for the dialog box and forms.

To customize the validation message and labels

1. Use an XML editor to open the package_res.xml file.
2. Add the following code to the Script section:

```
<str key="msg_channel_partner_present">This channel partner is already present in
the list.</str> <str key="msg_channel_partner_is_primary">This channel partner is
primary. The primary record cannot be removed. To remove this record, make another
record primary first.</str> <str key="msg_channel_partner_add_caption">Add
channel partner.</str><str key="lbl_channel_partner">Lead Partner Name</str>
```

3. Add the same code that you added in Step 2 to the Messages section.
4. Close and then save the package_res.xml file.

5. The following code specifies the values for labels that you use in the dialog box and form layouts:

   ```
   <str key="lbl_channel_partner">Lead Partner Name</str>
   ```

Publishing and Testing a Custom MVG Field

This task is a step in *Roadmap for Adding an MVG Field*.

In this topic, you publish and test your customization.

To publish and test a custom MVG field

1. Publish your changes.
   For more information, see *Publishing or Unpublishing Customization Packages*.
2. Test your changes:
   a. Open Oracle CRM On Demand Desktop, and then navigate to the Opportunity form.
   b. Verify that the form includes an MVG for the Lead Partner Name field.
   c. Click the MVG that Oracle CRM On Demand Desktop displays next to the Lead Partner Name field, and then verify that Oracle CRM On Demand Desktop does the following:
      - Displays the Channel Partners MVG dialog box
      - Displays the list of partners in the Associated Channel Partners window of the dialog box
      - Includes a partner record in the Primary window
   d. Enter letters in the Enter Value to Find Record window.
   e. Verify that Oracle CRM On Demand Desktop automatically displays records in accordance with the letters you enter.
   f. To verify the SalesBook control, click the SalesBook icon, and verify that Oracle CRM On Demand Desktop displays the SalesBook dialog box and that this dialog box displays a list of channel partners.

Example Code You Use to Add an MVG

This topic describes some of the code you use to add an MVG in this example. It includes the following information:

- *Code to Add a Custom Object Type*
- *Code to Map a Custom Object*
- *Code to Configure the Synchronization of a Custom Object*
- *Code to Add a Lookup View*
- *Code to Add a View*
- *Code to Add a New Association*
- *Code to Define a Sales Book Control*

Code to Add a Custom Object Type

To add a custom object type, you add the following code to the od_meta_info.xml file. To assist with debugging, it is recommended that you place this code after the last Type definition:

```
<object TypeId='ChannelPartner' Label='Channel Partner' LabelPlural='Channel
Partners' EnableGetIDsBatching='true' ViewMode='Sales Rep' IntObjName='Channel
Partner' SiebMsgXmlElemName='ChannelPartner'
SiebMsgXmlCollectionElemName='ListOfChannelPartner' >
 <field Name='DS Updated' Label='DS Updated' DataType='DTYPE_DATETIME'
 IsFilterable='no' IsHidden='yes' IOElemName='DSUpdated' />
 <field Name='Id' Label='Id' IsPrimaryKey='yes' DataType='DTYPE_ID'
 IsFilterable='no' IsHidden='yes' IOElemName='Id'/>
 <field Name='Name' Label='Name' DataType='DTYPE_TEXT' IsPartOfUserKey='yes'
 IOElemName='Name' />
 <field Name='Location' Label='Location' DataType='DTYPE_TEXT'
 IsPartOfUserKey='yes' IOElemName='Location'/>
</object>
```

This code does the following:

- Uses properties of the integration object and integration component as the values for attributes.
- References only a few of the many fields that exist in the Channel Partner object in Oracle CRM On Demand.
- Uses the Name and Location fields as parts of the user key.

  You define the natural key in the connector_configuration.xml file.

Code to Map a Custom Object

You can map a field of an Oracle CRM On Demand object to a Microsoft Outlook field or to a custom Oracle CRM On Demand Desktop field. For example, you can do the following:

- Map the Name field in Oracle CRM On Demand to the LastName field in Microsoft Outlook
- Map the mapLocation field in Microsoft Outlook to the Location field in Oracle CRM On Demand

To map objects, you add the following code to the od_basic_mapping.xml file:

```
<type id="Currency" icon="type_image:Currency:16">
 <field id ="Name">
 <type>
 <simple type="string"/>
 </type>
 </field>
 <field id ="Currency Symbol">
 <type>
 <simple type="string"/>
 </type>
 </field>
</type>
```

Note the following requirements:

- The value for the type id element in the od_basic_mapping.xml file must equal the value in the TypeId object in the od_meta_info.xml file.
- If the object you are adding is to be placed into DB FACADE storage, then the code must be as follows:

```
<type id="Currency" icon="type_image:Currency:16">
 <field id="Name">
 <type>
 <simple type="string" />
 </type>
 </field>
</type>
```

Code to Configure the Synchronization of a Custom Object

To configure synchronization for a custom object, you add the following code to the connector_configuration.xml file:

```
<type id="ChannelPartner">
 <view label="Channel Partner" label_plural="Channel Partners"
 small_icon="type_image:Account:16" normal_icon="type_image:Account:24"
 large_icon="type_image:Account:48"></view>
 <synchronizer name_format=":[:(Name):]">
 <links></links>
 <natural_keys>
 <natural_key>
 <field>Name</field>
 <field>Location</field>
 </natural_key>
 </natural_keys>
 </synchronizer>
</type>
```

Note the following:

- The value in the type id element must equal the value in the TypeId object in the meta_info.xml file.
- The natural_key element includes the Name and Location fields as part of the user key.

Code to Add a Lookup View

To add a lookup view, add the following code to the lookup_view_defs.xml file:

```
<lookup_view_def key="lookup:channel_partners">
 <display name="Channel Partners"></display>
 <filter dasl="[http://schemas.microsoft.com/mapi/proptag/0x001A001E] &gt;=
 'IPM.Contact.OnDemand.Channel_Partner' AND [http://schemas.microsoft.com/mapi/
 proptag/0x001A001E] &lt;= 'IPM.Contact.OnDemand.Channel_Partner'"></filter>
 <view id="channel_partner:salesbook"></view>
 <quick_lookup dasl_format="[http://schemas.microsoft.com/mapi/id/{00062004-0000-
 0000-C000-000000000046}/8005001E] = '%s'"></quick_lookup>
 <type id=""></type>
</lookup_view_def>
```

The following table describes important attributes you use in this code.

Attribute	Description
key	Id of the lookup control.
display name	Caption of the lookup control.
filter	Object that Oracle CRM On Demand Desktop displays on the lookup control. To specify an object type, use the message_type attribute from the od_basic_mapping.xml file.
view	Id of the salesbook control that Oracle CRM On Demand Desktop uses for this lookup.
type id	The type of object that Oracle CRM On Demand Desktop creates if the user clicks New on the lookup control. If this attribute is empty, then Oracle CRM On Demand Desktop disables the button.

Code to Add a View

To add the definition for a view, add the following code to the views.xml file:

```
<str key="channel_partner:mvg"> <![CDATA[<?xml version="1.0"?>
 <view>
 <viewname>Phone List</viewname>
 <viewstyle>table-layout:fixed;width:100%;font-family:Segoe UI;font-
 style:normal;font-weight:normal;font-size:8pt;color:Black;font-charset:0</
 viewstyle>
 <viewtime>0</viewtime>
 <linecolor>8421504</linecolor>
 <linestyle>3</linestyle> <gridlines>1</gridlines>
 <collapsestate/>
 <rowstyle>background-color:window;color:windowtext</rowstyle>
 <headerstyle>background-color:#D3D3D3</headerstyle>
 <previewstyle/>
 <arrangement>
 <autogroup>0</autogroup>
 <collapseclient/>
 </arrangement>
 <multiline>
 <width>0</width>
 </multiline>
 <column>
 <name>HREF</name>
 <prop>DAV:href</prop>
 <checkbox>1</checkbox>
 </column>
 <column>
 <type>string</type>
 <heading>Name</heading>
 <prop>http://schemas.microsoft.com/mapi/string/{00020329-0000-0000-C000-
 000000000046}/od%20PartnerName</prop>
 <width>426</width>
 <style>padding-left:3px;;text-align:left</style>
 <editable>1</editable>
 <userheading>Name</userheading>
 </column>
 <column>
 <type>string</type>
 <heading>Location</heading>
 <prop>http://schemas.microsoft.com/mapi/string/{00020329-0000-0000-C000-
 000000000046}/od%20PartnerLocation</prop>
 <width>426</width>
 <style>padding-left:3px;;text-align:left</style>
 <editable>1</editable>
 <userheading>Location</userheading>
 </column>
 <orderby>
 <order>
 <heading>File As</heading>
 <prop>urn:schemas:contacts:fileas</prop>
 <type>string</type>
 <sort>asc</sort>
 </order>
 </orderby>
 <groupbydefault>0</groupbydefault>
 <previewpane>
 <markasread>0</markasread>
 </previewpane>
 </view>]]>
</str>
```

Code to Add a New Association

To add a new association, add the following code to the business_logic.js file:

```
{
var opportunity_channel_partner =
 add_mvg_link("Opportunity", "ChannelPartner", "Primary Partner Id", null,
 "Opportunity.Channel_Partner.Association", "OpportunityId",
 "ChannelPartnerId", null, "ChannelPartnerStatus",
 null, ["lookup:channel_partners"],
 false, false, false, true);
 deny_primary_delete(opportunity_channel_partner.mvg1);//optional
 opportunity_channel_partner.mvg1.dialog_template_params =
 "dialog_caption": "#obj_activity_employee_plural",
 "autocomplete_display_format": ":[:(Name):]",
 "associations_view_caption": "#head_associated_channel",
 "associations_view_id": "channel_partner:mvg",
 "primary_selector_display_format": ":[:(PartnerName):]"
}
```

The ChannelPartnerStatus field contains the status of the Channel Partner object, such as unsaved, deleted, and so on.

The following table describes the important attributes you can use with the add_mvg_link function.

Attribute	Description
left_type	The type of the first linked object, for example, Opportunity.
right_type	The type of the second linked object, for example, Channel Partner.
left_obj_primary	The field of the Opportunity object. This field contains the primary Id for the Channel Partner object.
right_obj_primary	The field of the Channel Partner object. This field contains the primary Id for the Opportunity object.
assoc_type	The Id of the association type described in the od_meta_info.xml file and the od_basic_mapping.xml file.
left_link	The field of the association object that contains the Id of the opportunity.
right_link	The field of the association object that contains the Id of the channel partner.
left_assoc_status	The field of the association that contains the status of the opportunity.
right_assoc_status	The field of the association that contains the status of the channel partner.
left_objs_view_ids	The list of lookup view definitions you must use to display opportunity objects.
right_objs_view_ids	The list of lookup view definitions you must use to display channel partner objects.
left_primary_refresh _required	If the primary object is changed, and if the left_primary_refresh_required attribute is true, then Oracle CRM On Demand Desktop updates the object Id for the primary field of the opportunity that you specify in the left_obj_primary attribute.

Attribute	Description
	If you use the **writer class="link_fields"** element in the od_basic_mapping.xml file for this field, then set the left_primary_refresh_required attribute to True.
right_primary_refresh_required	If the primary object is changed, and if the right_primary_refresh_required attribute is true, then Oracle CRM On Demand Desktop updates the object Id for the primary field of the channel partner that you specify in the right_obj_primary attribute. If you use the **writer class="link_fields"** element in the od_basic_mapping.xml file for this field, then set the right_primary_refresh_required attribute to True.
assoc_left_link_refresh_required	If the opportunity object is changed, then Oracle CRM On Demand Desktop updates the OpportunityId field of the association that you specify in the left_link attribute. If you use the **writer class="link_fields"** element in the od_basic_mapping.xml file for this field, then set the assoc_left_link_refresh_required attribute to True.
assoc_right_link_refresh_required	If the channel partner object is changed, then Oracle CRM On Demand Desktop updates the ChannelPartnerId field of the association that you specify in the right_link attribute. If you use the **writer class="link_fields"** element in the od_basic_mapping.xml file for this field, then set the assoc_right_link_refresh_required attribute to True.

Code to Define a Sales Book Control

To define a SalesBook control, add the following code to the views.xml file.

```
<str key="channel_partner:salesbook">
 <![CDATA[<?xml version="1.0"?>
 <view>
 <viewname>Phone List</viewname>
 <viewstyle>table-layout:fixed;width:100%;font-family:Segoe UI;font-
style:normal;font-weight:normal;font-size:8pt;color:Black;font-charset:0</
viewstyle>
 <viewtime>0</viewtime>
 <linecolor>8421504</linecolor>
 <linestyle>3</linestyle>
 <gridlines>1</gridlines>
 <newitemrow>1</newitemrow>
 <collapsestate/>
 <rowstyle>background-color:window;color:windowtext</rowstyle>
 <headerstyle>background-color:#D3D3D3</headerstyle>
 <previewstyle/>
 <arrangement>
 <autogroup>0</autogroup>
 <collapseclient/>
 </arrangement>
 <multiline>
 <width>0</width>
 </multiline>
 <column>
 <name>HREF</name>
 <prop>DAV:href</prop>
 <checkbox>1</checkbox>
 </column>
 <column>
 <heading>Last Name</heading>
 <prop>urn:schemas:contacts:sn</prop>
 <type>string</type>
 <width>322</width>
```

```
    <style>padding-left:3px;;text-align:left</style>
    <editable>1</editable>
    </column>
    <column>
    <type>string</type>
    <heading>Location</heading>
    <prop>http://schemas.microsoft.com/mapi/string/{00020329-0000-0000-C000-
    000000000046}/od%20Location</prop>
    <width>322</width>
    <style>padding-left:3px;;text-align:left</style>
    <editable>1</editable>
    <userheading>Location</userheading>
    </column>
    <orderby>
    <order>
    <heading>File As</heading>
    <prop>urn:schemas:contacts:fileas</prop>
    <type>string</type>
    <sort>asc</sort>
    </order>
    </orderby>
    <groupbydefault>0</groupbydefault>
    <previewpane>
    <markasread>0</markasread>
    </previewpane>
    </view>]]>
  </str>
```

Setting Up Books for a Customized Package from Version 2.0

To set up book functionality from a customized Oracle CRM On Demand Desktop Version 2.0 package to work with the current version of Oracle CRM On Demand Desktop, complete the following procedure.

Before You Begin

Complete the steps in *Updating Oracle CRM On Demand Desktop Product-Specific Files from Version 5.0 to Version 5.1*.

To set up books for a customized package from Version 2.0

1. In the package_res.xml file, insert the following lines inside the `<res_root>` element:

```
<str key="lbl_book_of_business">Book Name:</str>
<str key="btn_book_select">...</str>
<str key="obj_book_plural">Books</str>
<str key="lbl_record_set_caption">Record Set</str>
<str key="cbx_record_set_team">All records where I am owner or on the team</str>
<str key="cbx_record_set_book">All records in a book</str>
<str key="cbx_record_set_default_book">All records in my default book</str>
<str key="btn_book_select">...</str>
<str key="head_record_type">Record Type</str>
<str key="head_visibility_model">Record Set</str>
<str key="head_book_name">Book Name</str>
<str key="record_set_tab_description">Please specify the visibility model to be used
for each record type. If using visibility based on Book of Business, you will need
to pick a book using the Book Selector. If a visibility mode is not specified for a
record type, then "Owner and Team" visibility will be used.</str>
```

```
<str key="caption_book_dlg">Select Book</str>
<str key="msg_book_is_required">Book Name is required.</str>
<str key="msg_book_cannot_contain_data">This book cannot contain data. Select
another one to proceed.</str>
<str key="msg_book_must_be_custom">Record must be associated to a custom book.</str>
<str key="msg_owner_must_be_empty">Owner must be empty.</str>
<str key="inaccessible_book">&lt;Inaccessible book&gt;</str>
<str key="lbl_include_book_subitems">Include Sub-Items</str>
<str key="book_selector_top_node">$obj_book_plural$</str>
<str key="msg_ownership_log" skip_translation="yes">Ownership mode of {0} is {1}</
str>
<str key="msg_Specifying_both_Book_and_Owner_is_prohibited">Specifying both Book
and Owner is prohibited.</str>
<str key="lbl_books_not_supported">There are no Books of Business defined for this
user. Only records where you are owner or a team member will be synchronized.</str>
<str key="books_retrieve_failed_title">Books of Business</str>
<str key="btn_clear">Clear</str>
<str key="filter_page-page_description">Please, specify the criteria you'd like to
use for determining which $remote_app_name$ items have to be synchronized with
$app_name$. The following filters are applied to selected categories and sub-
categories based on the data selected in the $lbl_record_set_caption$ tab.</str>
<str key="msg_subitems_required_for_book">The book you have selected is a parent
book and cannot contain data.
To synchronize using the related sub-books, enable "Include Sub-Books" to continue.
If this parent book has no sub-books, select another book.</str>
<str key="msg_cannot_select_book_subitems_disallowed">You cannot select a book that
cannot contain data.</str>
<str key="head_contact_book_info">Contact Book info</str>
<str key="head_account_book_info">Account Book info</str>
<str key="head_opportunity_book_info">Opportunity Book info</str>
<str key="head_lead_book_info">Lead Book info</str>
<str key="err_msg_invalid_request">Request contains invalid data</str>
<str key="msg_book_access_denied">You don't have access rights to Books of Business,
used for synchronization or the Books are no longer valid. Please check your Books
of Business configuration settings in the Control Panel or contact technical support
for assistance.</str>
<str key="msg_record_access_denied">Access denied.</str>
<str key="msg_record_set_query_save_record_set">Do you want to save changes to
record set?</str>
```

2. In the dialogs.xml file, make the following updates:

   a. Remove the following lines from the <dialogs> element:

```
<dialog id="Control panel">
 <script>
 </script>
 <layout caption="#control_panel_title" sizable="true"
 small_icon="app_small_icon">
 <appearance height="480" width="744" position="desktop_center" />
 <cell>
 <stack layout="vert" rollup="true">
 <cell size="70">
 <cp_toolbar id="toolbar"/>
 </cell>
 <cell size="32" autosize="true">
 <cp_infobar id="infobar"/>
 </cell>
 <cell>
 <pages id="cp_pages" tab_order="2">
 <page id="cp_busy_page" layout="cp_busy_page"/>
 <page id="cp_filters_page" layout="cp_filters_page"/>
 <page id="cp_issues_page" layout="cp_issues_page"/>
 <page id="cp_conflicts_page" layout="cp_conflicts_page"/>
 <page id="cp_duplicates_page" layout="cp_duplicates_page"/>
 <page id="cp_confirmations_page" layout="cp_confirmations_page"/>
 </pages>
 </cell>
 </stack>
 </cell>
 </layout>
</dialog>
```

b. Insert the following lines inside the <dialogs> element:

```
<dialog id="select_book_dialog">
 <script><![CDATA[]]>
 </script>
 <layout sizable="true" id="General" caption="#caption_book_dlg"
 min_height="300" min_width="600" small_icon="app_small_icon">
 <appearance height="300" width="600" position="desktop_center" />
 <cell>
 <stack layout="horz" padding="10">
 <cell>
 <stack layout="vert" padding="10" spacing="5">
 <cell>
 <stack layout="horz" spacing="5" padding="5">
 <cell>
 <treeview id="books_tree" tab_order="1" lines_at_root="true"/>
 </cell>
 </stack>
 </cell>
 <cell size="10" />
 <cell size="25">
 <stack layout="horz" spacing="5">
 <cell size="75">
 <button tab_order="2" id="btn_clear" visible="false">
 <text>#btn_clear</text>
 </button>
 </cell>
 <cell>
 <checkbox id="include_subitems" tab_order="3" visible="false">
 <text>#lbl_include_book_subitems</text>
 </checkbox>
 </cell>
 <cell />
 <cell size="75">
 <button tab_order="4" id="btn_ok">
 <text>#btn_ok</text>
 </button>
 </cell>
 <cell size="75">
 <button id="btn_cancel" tab_order="5">
 <text>#btn_cancel</text>
 </button>
 </cell>
 </stack>
 </cell>
 </stack>
 </cell>
 </stack>
 </cell>
 </layout>
</dialog>
```

3. In the connector_configuration.xml file, make the following updates:

a. Locate the <bindings> section and make the following updates:

- Add the following code to <type name="Opportunity"> section:

```
<binding>
 <field name="OwnerId"/>
 <field name="BookId"/>
</binding>
```

- Add the following code to the <type name="Activity"> section:

```
<binding>
 <field name="OwnerId"/>
 <field name="BookId"/>
</binding>
```

- Add the following code to <bindings>:

```
<type name="Contact">
 <binding>
 <field name="OwnerId"/>
 <field name="BookId"/>
 </binding>
</type>
<type name="Account">
 <binding>
 <field name="OwnerId"/>
 <field name="BookId"/>
 </binding>
</type>
<type name="Lead">
 <binding>
 <field name="OwnerId"/>
 <field name="BookId"/>
 </binding>
</type>
```

b. Add the following code to the <types> section:

```
<type id="Book.Type.Association"> <!--Fake type to store user selected data-->
 <view label="Book.Type.Association" label_plural="Book.Type.Association"
 small_icon="type_image:Generic:16" normal_icon="type_image:Generic:24"
 large_icon="type_image:Generic:48" suppress_sync_ui="true"></view>
 <synchronizer sync_it="false" name_format=":[:(BookId) :]:[:(TypeName):]"
 threshold="0">
 <natural_keys>
 <natural_key>
 <field>TypeName</field>
 </natural_key>
 </natural_keys>
 </synchronizer>
</type>
<type id="Book">
 <view label="Book" label_plural="Books" small_icon="type_image:Generic:16"
 normal_icon="type_image:Generic:24" large_icon="type_image:Generic:48"
 suppress_on_top="true" suppress_sync_ui="true"></view>
 <synchronizer name_format=":[:(Label):]" threshold="0" frequency="0">
 <links></links>
 </synchronizer>
</type>
<type id="BookDefaults">
 <view label="BookDefaults" label_plural="BookDefaults"
 small_icon="type_image:Generic:16" normal_icon="type_image:Generic:24"
 large_icon="type_image:Generic:48" suppress_sync_ui="true"></view>
 <synchronizer name_format=":[:(Value) :]:[:(BookId):]" threshold="0"
 frequency="0">
 <links></links>
 </synchronizer>
```

```
</type>
<type id="RequiredFields">
 <view label="RequiredFields" label_plural="RequiredFields"
 small_icon="type_image:Generic:16" normal_icon="type_image:Generic:24"
 large_icon="type_image:Generic:48" suppress_sync_ui="true"></view>
 <synchronizer name_format=":[:(TypeName) :]:[:(Value):]" threshold="0"
 frequency="0">
 <links></links>
 </synchronizer>
</type>
```

c. Replace the two occurrences of the `<field>OwnerId</field>` element inside the <`type id="Activity">` `<natural_keys>` section with the following line:

```
<field allow_null="true">OwnerId</field>
```

4. Update the data_sources.xml file, as follows:

   a. Replace all occurrences of the `scope="general"` attribute with the following value:

```
scope="6001"
```

   b. Insert the following lines inside the <connector> element:

```
<data_source name="Book.Type.Association">
 <type>Book.Type.Association</type>
 <columns>
 <column>TypeName</column>
 <column>BookId</column>
 <column>IncludeSubBooks</column>
 </columns>
</data_source>
<data_source name="::Defaults">
 <type>::Defaults</type>
 <columns>
 <column>DefaultBookId</column>
 <column>DefaultBookName</column>
 </columns>
</data_source>
<data_source name="Book">
 <type>Book</type>
 <columns>
 <column>Label</column>
 <column>Value</column>
 <column>ParentBookName</column>
 <column>ParentBookId</column>
 <column>BookType</column>
 <column>CanContainDataFlag</column>
 </columns>
</data_source>
<data_source name="BookDefaults">
 <type>BookDefaults</type>
 <columns>
 <column>Label</column>
 <column>Value</column>
 <column>BookId</column>
 <column>IncludeSubBooks</column>
 <column>OrderNumber</column>
 </columns>
</data_source>
```

5. In the forms_12.xml files, add a book selector control and a label to each form for the book-enabled object types, and provide a tab_order attribute value for those elements:

- Label:

```
<static id="lbl_book_of_business" tab_order="…">
 <text>#lbl_book_of_business</text>
</static>
```

- Book selector control:

```
<stack layout="horz">
 <cell>
 <scriptable_edit id="book_of_business" tab_order="…">
 <field value="string"></field>
 </scriptable_edit>
 </cell>
 <cell size="5"></cell>
 <cell size="22">
 <button id="btn_book_select" tab_order="…">
 <text>#btn_book_select</text>
 </button>
 </cell>
</stack>
```

6. Update the od_basic_mapping.xml file as follows:

   a. Insert the following lines inside the `<type id = "User">` element:

```
<field id = "PrimaryPositionId"> <!-- Custom Book id -->
 <type>
 <simple type = "string"/>
 </type>
</field>
```

   b. Insert the following lines inside the `<type id = " ::Defaults">` element:

```
<field id = "DefaultBookId">
 <type>
 <simple type = "string"/>
 </type>
</field>
<field id = "DefaultBookName">
 <type>
 <simple type = "string"/>
 </type>
</field>
<field id = "Book">
 <type>
 <simple type = "string"/>
 </type>
</field>
<field id = "TypeDefaultBook">
 <type>
 <simple type = "string"/>
 </type>
</field>
<field id = "RequiredFields">
 <type>
 <simple type = "string"/>
 </type>
</field>
<field id = "PrimaryPositionId">
 <type>
 <simple type = "string"/>
 </type>
```

```
</field>
<field id = "FullName">
 <type>
 <simple type = "string"/>
 </type>
</field>
```

c. Add new types by inserting XML code inside the `<database> <types>` element. For a listing of the code to add, see *XML Code Addition for the od_basic_mapping.xml File*.

d. Add the BookId field to each book-enabled object type as follows:

```
<field id="BookId">
 <reader>
 <mapi_user>
 <user_field id="od BookId" ol_field_type="1"></user_field>
 <convertor>
 <string/>
 </convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="od BookId" ol_field_type="1"></user_field>
 <convertor>
 <string/>
 </convertor>
 </outlook_user>
 </writer>
</field>
```

7. Update the od_meta_info.xml file as follows:

a. Locate the following lines:

```
<product_flavor>OD</product_flavor>
<error_codes Type="RecordAlreadyExists" >SBL-EAI-04383/SBL-EAI-04383, SBL-
DAT-00381/SBL-EAI-04381, SBL-DAT-00357/SBL-EAI-04375, SBL-DAT-00381/SBL-DAT-
00381, SBL-DAT-00357/SBL-DAT-00357, SBL-DAT-00215/SBL-DAT-00215</
error_codes>
<error_codes Type="RecordAlreadyDeleted" >SBL-EAI-04378/SBL-EAI-04378</
error_codes>
<error_codes Type="AccessDenied" >*/SBL-ODS-50085</error_codes>
<error_codes Type="SessionExpired" >SBL-ODU-01006/*</error_codes>
<error_codes Type="RecordUpdatedPrematurely" >SBL-DAT-00523/SBL-EAI-04290</
error_codes>
```

b. Replace these lines with the following:

```
<errors>
 <codes Type="RecordAlreadyExists">SBL-EAI-04383/SBL-EAI-04383, SBL-DAT-
 00381/SBL-EAI-04381, SBL-DAT-00357/SBL-EAI-04375, SBL-DAT-00381/SBL-
 DAT-00381, SBL-DAT-00357/SBL-DAT-00357, SBL-DAT-00215/SBL-DAT-00215</
 codes>
 <codes Type="RecordAlreadyDeleted">SBL-EAI-04378/SBL-EAI-04378</codes>
 <codes Type="AccessDenied">*/SBL-ODS-50085</codes>
 <codes Type="SessionExpired">SBL-ODU-01006/*</codes>
 <codes Type="RecordUpdatedPrematurely">SBL-DAT-00523/SBL-EAI-04290</
 codes>
 <codes Type="BookAccessDenied"
 ErrMsgReplacement="#msg_book_access_denied" PassToBalloon="true">SBL-
 EAI-04471/SBL-EAI-04471</codes>
 <codes Type="RecordAccessDenied"
 ErrMsgReplacement="#msg_record_access_denied">SBL-DAT-00542/SBL-EAI-
 04376</codes>
```

```
</errors>
```

c. Locate the following line:

```
<object TypeId="::Defaults" Label="#obj_defaults" LabelPlural="#obj_defaults"
IsAssociation="no" IsTopLevel="yes" AllowTypedefCaching="false">
```

d. Replace this line with the following:

```
<object TypeId="::Defaults" Label="#obj_defaults" LabelPlural="#obj_defaults"
IsAssociation="no" IsTopLevel="yes">
```

e. Insert the following lines inside the `<common_settings>` element:

```
<access_modes_config>
 <books Enable="true" >
 <request_scopes>
 <scope Name="General" />
 <scope Name="QBID" />
 </request_scopes>

 <t2b_bnd_config StorageTname="Book.Type.Association"
 TnameFldname="TypeName" BookIdFldName="BookId"
 SubbooksFldname="IncludeSubBooks" />
 <default_books_picklist Name="TypeDefaultBookPL" />
 </books>
</access_modes_config>
```

f. Locate the following line:

```
<field Name="Id" Label="Id" IsPrimaryKey="yes" IsReadonly="yes"
IsRequired="no" DataType="DTYPE_ID" IsNullable="no" />
```

g. Replace this line with the following:

```
<field Name="RawId" Label="RawId" IsPrimaryKey="yes" IsReadonly="yes"
DataType="DTYPE_ID" IsNullable="no" IsHidden="true" IsFake="true" />
```

h. Add the following fields to the `<object TypeId="::Defaults">` element:

```
<field Name="FullName" Label="FullName" DataType="DTYPE_TEXT" />
<field Name="PrimaryPositionId" Label="PrimaryPositionId"
DataType="DTYPE_TEXT" />
<field Name="Book" Label="Book" DataType="DTYPE_TEXT" IsHidden="true"
IsFake="true" PicklistTypeId="BookPL" />
<field Name="DefaultBookId" Label="DefaultBookId" DataType="DTYPE_TEXT" />
<field Name="DefaultBookName" Label="DefaultBookName" DataType="DTYPE_TEXT" /
>
<field Name="TypeDefaultBook" Label="TypeDefaultBook" DataType="DTYPE_TEXT"
IsHidden="true" IsFake="true" PicklistTypeId="TypeDefaultBookPL" />
<field Name="RequiredFields" Label="RequiredFields" DataType="DTYPE_TEXT"
IsHidden="true" IsFake="true" PicklistTypeId="FieldsInfoPL" />
```

i. For each `<object>` element which contains the `<extra_command_options>` element, do the following:

  - Locate the following line:

    ```
    <option Name="UseDefaultViewMode" Value="true" Scopes="Dedup" />
    ```

  - Replace this line with the following:

    ```
    <option Name="UseDefaultViewMode" Value="true" Scopes="Dedup;ONLKP;QBID" />
    ```

- Insert the following lines inside the `<object>` element:

```
<viewmodes>
 <viewmode Name="None" Scopes="Dedup;ONLKP;QBID" />
</viewmodes>
```

j. For each `<object>` element of a book-enabled object type, do the following:

- Insert the following lines inside the `<object>` element, where `TypeOrderNumber` is the order number of the Type in the connector_configuretion.xml file:

```
<access_mode_config BooksSupported="true" TypeOrderNumber="5" >
 <involved_fields>
 <field Name="Owner" />
 <field Name="BookName" />
 </involved_fields>
 <scopes>
 <scope Id="Dedup" BookId="" />
 <scope Id="ONLKP" BookId="" />
 <scope Id="QBID" BookId="" />
 </scopes>
</access_mode_config>
```

- Add the following fields to the `<object>` element of the Contact, Account, Opportunity, and Lead record types:

```
<field Name="BookId" Label="BookId" DataType="DTYPE_TEXT" CRMName="Primary
Position Id" />
<field Name="BookName" Label="BookName" DataType="DTYPE_TEXT"
CRMName="Primary Position Name" >
 <suppress_on Upstream="true" />
</field>
```

- Add the following fields to the `<object>` element of the Activity record type:

```
<field Name="BookId" Label="BookId" DataType="DTYPE_TEXT" CRMName="Position
Id" />
<field Name="BookName" Label="BookName" DataType="DTYPE_TEXT"
CRMName="Position Name" >
 <suppress_on Upstream="true" />
</field>
```

- Replace the `<field Name='Owner' … />` element of the Contact, Account, Opportunity, and Lead record types with the following field element:

```
<field Name='Owner' Label='Owner' DataType='DTYPE_TEXT' IsFilterable='no'
IsHidden="yes" CRMName="Owner Alias" >
 <suppress_on Upstream="true" />
</field>
```

- Add the following field element to the Activity object:

```
<field Name="Owner" Label="Owner" DataType="DTYPE_ID" IsFake="true"
IsFilterable='no' CRMName="Assigned To" IsHidden="yes">
 <suppress_on Upstream="true" />
</field>
```

k. Inside the `<root>` element, add the lines of XML code listed in *XML Code Addition for the od_meta_info.xml File*.

8. In the od_helpers.js file, after the first line of the file, insert the JavaScript code listed in *JavaScript Code Addition for the od_helpers.js File*.

9. Update the application_script.js file as follows:

   a. Insert the following code line after the **`var cp_pages = [`** line:

      ```
      {native: false, layout: "cp_record_set_page", available: true, image:
      "type_image:Book:32", caption:
      application.session.res_string("lbl_record_set_caption")},
      ```

   b. Locate the existing **`fra_handler(fra)`** function:

      ```
      function fra_handler(fra)
      {
       var current_form = null;
       var on_closed = function()
      {
       current_form = null;
       fra.exit_current_step(false);
       }
       function on_fra_step(id)
       {

       if (id == "advanced")
       {
       var xml = ui.get_dialog_xml("PropSheetHost");
       xml = helpers.replace_all("$prop_sheet_layout",
       "options_advanced_page", xml);
       current_form = ui.create_dialog_from_xml(0, xml);
       current_form.on_closed.connect(on_closed)
       current_form.visible = true;
       }
       }
       fra.on_step.connect(on_fra_step);
       fra.add_builtin_step("welcome");
       fra.add_builtin_step("sync_filters");
       fra.add_builtin_step("sync_schedule");
       fra.add_step("advanced", session.res_string("sa-advanced_settings-
       caption"), session.res_string("sa-advanced_settings-description"), "sa-
       advanced_settings-picture", true);
       fra.add_builtin_step("convert_items");
       fra.add_builtin_step("first_sync");
       }
      ```

   c. Replace the existing **`fra_handler()`** function with the following:

      ```
      function fra_handler(fra)
      {
       var current_form = null;

       var on_closed = function()
       {
       current_form = null;
       fra.exit_current_step(false);
       }

       function on_fra_step(id)
       {
       switch (id) {
       case "advanced": step_advanced(); break;
       case "record_set": step_record_set(); break;
       case "server_configuration": step_server_configuration(); break;
       }
      ```

```
 function step_advanced() {
 var xml = ui.get_dialog_xml("PropSheetHost");
 xml = helpers.replace_all("$prop_sheet_layout", "options_advanced_page",
 xml);
 current_form = ui.create_dialog_from_xml(0, xml);
 current_form.on_closed.connect(on_closed);
 current_form.visible = true;
 }

 function step_record_set() {
 var cp_ctx =
 {
 "open_context": "cp_record_set_page",
 "single_page": true,
 "on_closed_fn": on_closed
 };
 sync_ui_session..show_cp(cp_ctx);
 }

 function step_server_configuration() {
 current_form = ui.create_dialog(0, "FRA Server Configuration Dlg");
 current_form.on_closed.connect(on_closed);
 current_form.visible = true;
 }
 }

 fra.on_step.connect(on_fra_step);
 fra.add_builtin_step("welcome");
 fra.add_step("server_configuration", application.session.res_string("sa-
 server_configuration-caption"), application.session.res_string("sa-
 server_configuration-description"), "sa-server_configuration-picture", true);
 fra.add_step("record_set", application.session.res_string("sa-record_set-
 caption"), application.session.res_string("sa-record_set-description"), "sa-
 record_set-picture", true);
 fra.add_builtin_step("sync_filters");
 fra.add_builtin_step("sync_schedule");
 fra.add_step("advanced", application.session.res_string("sa-
 advanced_settings-caption"), application.session.res_string("sa-
 advanced_settings-description"), "sa-advanced_settings-picture", true);
 fra.add_builtin_step("convert_items");
 fra.add_builtin_step("first_sync");
}
```

d. Add the following code to the end of the file:

```
var ownership_logger = application.create_state_set();
var types_to_log = ["Activity", "Contact", "Opportunity", "Lead", "Account"];
ownership_logger.add("synchronizing", false);
ownership_logger.on_changed.connect(function(synchronizing_finished){
 if(synchronizing_finished)
 {
 var sync_is_finished = ownership_logger.is_actual;
 helpers.for_each2(types_to_log, function(type_id){
 application.logger.info(helpers.format(session.res_string("msg_owners
 hip_log"),[type_id, od_helpers.get_ownership_mode(g_ctx, type_id)]));
 });
 // Store Users' Custom Book Ids for autoresolver
 var book_ids = [];
 helpers.for_each(session.find_items("User", session.empty_criteria),
 function(item) { book_ids.push(item["PrimaryPositionId"]); });
 g_ctx.application.settings.set("CustomBookIds", book_ids.join(";"));
 }
});
```

10. Update the autoresolver.js file as follows:

a. Replace the substring **`left_values`** with **`remote_values`** throughout the entire file.

b. Replace the substring **`right_values`** with **`local_values`** throughout the entire file.

c. Add the following **`resolve_custom_book()`** function to the end of the file:

```
function resolve_custom_book(field_name, local_set, remote_set, resolution)
{
 var lid = local_set.get_field(field_name);
 var rid = remote_set.get_field(field_name);
 var custom_book_ids = application.settings.get("CustomBookIds");
 if (lid == "" && custom_book_ids != null && custom_book_ids.search(rid) > -1)
 return resolution.set_field(field_name, rid), true;
 return false;
}
```

d. Locate the following line:

```
resolve_to_winner("OwnerId", remote_set, local_set, resolution); //OLOD-284
```

e. Add the following after the line:

```
resolve_custom_book("BookId", local_set, remote_set, resolution);
```

f. Locate the following line:

```
resolve_to_local_nonempty("PrimaryShipToCounty", local_set, remote_set,
resolution);
```

g. Add the following after the line:

```
resolve_custom_book("BookId", local_set, remote_set, resolution);
```

h. Locate the following line:

```
resolve_to_local_nonempty("AlternateCounty", local_set, remote_set,
resolution)
```

i. Add the following after the line:

```
resolve_custom_book("BookId", local_set, remote_set, resolution);
```

j. Locate the following line:

```
resolve_zero_to_null("Revenue", local_set, remote_set, resolution)
```

k. Add the following after the line:

```
resolve_custom_book("BookId", local_set, remote_set, resolution);
```

l. Locate the following line:

```
resolve_to_local_nonempty("County", local_set, remote_set, resolution);
```

m. Add the following after the line:

```
resolve_custom_book("BookId", local_set, remote_set, resolution);
```

11. Update the business_logic.js file as follows:

    a. Locate the following line:

```
if(!(ctx.form.item.snapshot.AccountId != null && account_id == null))
```

b. Replace the line with the following:

```
if(!(action_ctx.item_ex.get_property("AccountId") != null && account_id ==
null))
```

c. Locate the following function:

```
function deny_primary_delete(mvg_link)
{
 scheme.triggers.add_simple_trigger(data_model.deny_primary_delete_trigger,
mvg_link.type, mvg_link.link_to, mvg_link.tag, "removing");
}
```

d. Replace the function with the following functions:

```
function deny_primary_delete(mvg_link)
{
 scheme.triggers.add_simple_trigger(deny_primary_delete_trigger_wrapper,
 mvg_link.type, mvg_link.link_to, mvg_link.tag, "removing");
 scheme.triggers.add_simple_trigger(deny_primary_set_non_current_user,
 mvg_link.type, mvg_link.link_to, null, "creating");
 function deny_primary_set_non_current_user(ctx, action_ctx)
 {
 var mode = od_helpers.get_ownership_mode(ctx, action_ctx.type);
 if(mode == "mixed" || mode == "owner")
 {
 var curr_user_id = helpers.get_current_user_id(ctx.session),
 with_user_id = action_ctx.with_id;
 var is_manual_operation = action_ctx.params == null ||
 action_ctx.params.is_manual_operation;
 // prevent manual set primary to non-current user
 if (is_manual_operation && with_user_id != null && action_ctx.tag ==
 "direct" && !helpers.ids_equal(ctx.session, curr_user_id,
 with_user_id )) action_ctx.notification_params.cancel = true;
 // set current user primary, allowed for manual operation only
 if (is_manual_operation && with_user_id != null && action_ctx.tag
 == "mvg" && helpers.ids_equal(ctx.session, curr_user_id,
 with_user_id ))
 {
 var linked_direct = ctx.linker.linked({ "item_ex":
 action_ctx.item_ex, "tag": "direct", "link_to":
 action_ctx.link_to });
 ctx.linker.link({ "item_ex": action_ctx.item_ex, "link_to":
 action_ctx.link_to, "tag": "direct", "with_id": with_user_id });
 var linked = ctx.linker.linked({ "item_ex": action_ctx.item_ex,
 "tag": "mvg", "link_to": action_ctx.link_to });
 if(linked_direct.value == null &&
 helpers.contains_object_id(ctx.session, linked.values,
 with_user_id)) // owner was empty but association with current user
 alrady exists
 action_ctx.notification_params.cancel = true; // prevent "already
 present" msg
 }
 }
 }
 function deny_primary_delete_trigger_wrapper(ctx, action_ctx)
 {
 var mode = od_helpers.get_ownership_mode(ctx, action_ctx.type);
 if(mode == "owner")
 data_model.deny_primary_delete_trigger(ctx, action_ctx);
 }
}
```

e. Locate the following function:

```
function prefill_owner(ctx)
{
 var current_user_id = helpers.get_current_user_id(ctx.session);
 if (current_user_id != null)
 {
 return ([
 { "with_id": current_user_id, "tag": "mvg" },
 { "with_id": current_user_id, "tag": "direct" }
 ]);
 }
 else return [];
}
```

f. Replace the function with the following code:

```
function prefill_owner(ctx, type_id)
{
 type_id = type_id || ctx.item_ex.get_type();
 var current_user_id = helpers.get_current_user_id(ctx.session);
 if (current_user_id != null)
 {
 var ownership_mode = od_helpers.get_ownership_mode(ctx, type_id);
 var fill_owner = ownership_mode != "book" && !(ownership_mode == "mixed"
 && prefill_book(ctx, type_id)); // prefill owner in mixed mode if no book
 was prefilled
 var links = [
 { "with_id": current_user_id, "tag": "mvg" }, // add to team
 { "with_id": current_user_id, "tag": "direct" } // fill owner
 ];
 if(!fill_owner)
 links = [];
 return links;
 }
 else return [];
}
function prefill_activity_owner(ctx)
{
 return prefill_owner(ctx, "Activity");
}
function prefill_book(ctx, type_id)
{
 type_id = type_id || ctx.item_ex.get_type();
 var ownership_mode = od_helpers.get_ownership_mode(ctx, type_id);
 switch (ownership_mode)
 {
 case "book":
 var user_selected_book = od_helpers.get_ol_storage_rs_book(ctx,
 type_id);
 var book_id = user_selected_book != null ? user_selected_book.BookId
 : "";
 if (!book_id) {
 var default_book = ctx.session.find_item("BookDefaults",
 ctx.session.create_expression("Value", "eq", type_id));
 book_id = default_book != null ? default_book.BookId : "";
 }
 if(!book_id)
 {
 var defaults = helpers.get_defaults(ctx.session),
 user_default_book_name = defaults != null ?
 defaults.DefaultBookName : "",
 user_default_book = ctx.session.find_item("Book",
 ctx.session.create_expression("Label", "eq",
 user_default_book_name));
 book_id = user_default_book != null ? user_default_book.Value :
 book_id;
 }
 if(od_helpers.book_can_contain_data(ctx, book_id))
 return book_id;
 break;
 case "mixed":
 var user_selected_book = od_helpers.get_ol_storage_rs_book(ctx,
 type_id);
 var book_id = user_selected_book != null ? user_selected_book.BookId
 : "";
 if (book_id && od_helpers.book_can_contain_data(ctx, book_id))
 return book_id;
 break;
 default:
 return "";
 break;
 }
}
function prefill_activity_book(ctx)

{
 return prefill_book(ctx, "Activity");
}
function prefill_lead_owner_od(ctx)
{
 var current_user_id = null;
 if(ctx.security_descriptor.ownership_mode() != "book")
 {
 current_user_id = helpers.get_current_user_id(ctx.session);
 }
 return current_user_id;}
}
```

g. Locate the following line:

```
scheme.objects.get_object("Activity")["initial_links_fn"] = prefill_owner;

scheme.objects.get_object("Activity").get_field("OwnerId")["initial_value_fn
"] = prefill_owner_od;
```

h. Replace the line with the following:

```
scheme.objects.get_object("Activity")["initial_links_fn"] =
prefill_activity_owner;

scheme.objects.get_object("Activity").get_field("BookId")["initial_value_fn"
] = prefill_activity_book;
```

i. Locate the following line:

```
scheme.objects.get_object("Contact").get_field("OwnerId")["initial_value_fn"
] = prefill_owner_od;
```

j. Replace the line with the following:

```
scheme.objects.get_object("Contact").get_field("BookId")["initial_value_fn"]
= prefill_book;
```

k. Locate the following line:

```
scheme.objects.get_object("Account").get_field("OwnerId")["initial_value_fn"
] = prefill_owner_od;
```

l. Reeplace the line with the following:

```
scheme.objects.get_object("Account").get_field("BookId")["initial_value_fn"]
= prefill_book;
```

m. Locate the following line:

```
scheme.objects.get_object("Lead").get_field("OwnerId")["initial_value_fn"] =
prefill_owner_od;
```

n. Replace the line with the following:

```
scheme.objects.get_object("Lead").get_field("OwnerId")["initial_value_fn"] =
prefill_lead_owner_od;

scheme.objects.get_object("Lead").get_field("BookId")["initial_value_fn"] =
prefill_book;
```

o. Locate the following line:

```
scheme.objects.get_object("Opportunity").get_field("OwnerId")["initial_value
_fn"] = prefill_owner_od;
```

p. Replace the line with the following:

```
scheme.objects.get_object("Opportunity").get_field("BookId")["initial_value_
fn"] = prefill_book;
```

q. Locate the following line:

```
add_association_description(owner_result.owner, proxy, null, "direct");
```

r. Replace the line with the following:

```
if(activity_ownership_mode() != "book" &&
!proxy.item_ex().get_property("BookId"))
 add_association_description(owner_result.owner, proxy, null, "direct");
```

s. Locate the following line:

```
var const_ctx = { "item_ex": proxy.item_ex(), "submited": submited != false };
```

t. Replace the line with the following:

```
var const_ctx = { "item_ex": proxy.item_ex(), "submited": submited != false,
"params": {"is_manual_operation": false} };
```

u. Locate the following line:

```
if(helpers.contains_object_id(ctx.session, proxy_team.values, current_user))
```

v. Replace the line with the following:

```
if(activity_ownership_mode() != "owner" ||
helpers.contains_object_id(ctx.session, proxy_team.values, current_user))
```

w. Locate the following lines:

```
// var owner_filter = ctx.session.create_expression("Primary Owner Id", "eq",
owner);
```

```
 var owner_filter = ctx.session.create_expression("OwnerId", "eq", owner);
```

x. Replace the lines with the following:

```
 var owner_filter = ctx.session.create_expression("OwnerId", "eq", owner),
 ownership_mode = activity_ownership_mode();
```

y. Locate the following line:

```
conversation_id_filter.add(owner_filter);
```

z. Replace the line with the following:

```
if(ownership_mode == "owner")
 conversation_id_filter.add(owner_filter);
```

aa. Locate the following line:

```
filter.add(owner_filter);
```

ab. Replace the line with the following:

```
if(ownership_mode == "owner")
 filter.add(owner_filter);
```

ac. Locate the following line:

```
if (helpers.ids_equal(ctx.session, owner.value,
get_owner_for_ol_item(ol).owner)
```

ad. Replace the line with the following:

```
if ((activity_ownership_mode() != "owner" || helpers.ids_equal(ctx.session,
owner.value, get_owner_for_ol_item(ol).owner))
```

ae. Locate the following line:

```
if (is_current_user(primary_employee.value) || object_is_not_sync)
```

af. Replace the line with the following:

```
if (activity_ownership_mode() == "owner" &&
is_current_user(primary_employee.value) || object_is_not_sync)
```

ag. Insert the following function inside the activity_processor(ctx) function:

```
function activity_ownership_mode()
{
 return od_helpers.get_ownership_mode(ctx, "Activity")
}
```

12. Update the forms.js file as follows:

a. Insert the `register_book_control(ctx, owner_ctrl)` function at the end of the file. For a listing of the code, see *JavaScript Code Addition for the forms.js File*.

b. Locate the following line inside the `activity_subform(ctx)` function:

```
activity_form_base.apply(this, arguments);
```

c. Add the following code after the line:

```
register_book_control(ctx, "ActionToEmployee");
```

d. Make the following changes inside the `activity_form(ctx)` function:

- Remove the following lines:

  ```
  // enabling autoprefilling this item;

  ctx.form_links_manager.init_new();
  ```

- Locate the following line:

  ```
  activity_form_base.apply(this, arguments);
  ```

- Add the following code after the line:

  ```
  // enabling autoprefilling this item;
  ctx.form_links_manager.init_new();
  register_book_control(ctx, "ActionToEmployee");
  ```

- Remove the following line:

  ```
  validator.validate_empty_field("OwnerId", "ActionToEmployee",
  "msg_activity_owner_validation", false);
  ```

e. Locate the following line inside the `od_contact_form(ctx)` function:

```
ctx.form_links_manager.init_new();
```

f. Add the following after the line:

```
register_book_control(ctx, "TeamToContact");
```

g. Locate the following line inside the `od_lead_form(ctx)` function:

```
ctx.form_links_manager.init_new();
```

h. Add the following after the line:

```
register_book_control(ctx);
```

i. Make the following changes inside the `od_opportunity_form(ctx)` function:

- Remove the following lines:

  ```
  // enabling autoprefilling this item;
  ctx.form_links_manager.init_new();
  ```

- Remove the following lines:

  ```
  Status_control_changed();
  if (ctx.item_ex.get_id() == null) {
   Type_control_changed();
  } else {
   update_picklistSalesStage_control(form.item.snapshot.SalesProcessId);
  }
  ```

- Locate the following line:

```
 ctx.form.on_saved.connect(form_saved);
```

- Add the following code after the line:

```
// enabling autoprefilling this item;
ctx.form_links_manager.init_new();

 register_book_control(ctx, "TeamToOpportunity");

 Status_control_changed();
if (ctx.item_ex.get_id() == null) {
 Type_control_changed();
} else {
 update_picklistSalesStage_control(form.item.snapshot.SalesProcessId);
}
```

j. Make the following changes inside the `od_opportunity_form(ctx)` function:

- Remove the following lines:

```
// enabling autoprefilling this item;

ctx.form_links_manager.init_new();
```

- Locate the following line:

```
 ctx.form.on_saved.connect(form_saved);
```

k. Add the following code after the line:

```
// enabling autoprefilling this item;
ctx.form_links_manager.init_new();

 register_book_control(ctx, "TeamToAccount");
```

11 How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Data and Microsoft Outlook Data

How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Data and Microsoft Outlook Data

This appendix describes how Oracle CRM On Demand Desktop maps fields between Oracle CRM On Demand data and Microsoft Outlook data. It includes the following topics:

- *How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Activities and Microsoft Outlook Calendar*
- *How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Activities and Microsoft Outlook Tasks*
- *How Oracle CRM On Demand Desktop Maps Fields Between an Oracle CRM On Demand Activity and a Microsoft Outlook Email*

How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Activities and Microsoft Outlook Calendar

The following table describes how Oracle CRM On Demand Desktop maps objects between an Oracle CRM On Demand activity and Microsoft Outlook calendar.

Oracle CRM On Demand Field	Microsoft Outlook Field	Required	Comments
Owner	For more information, see *How Oracle CRM On Demand Assigns the Meeting Organizer*.	Yes	None.
PIM Meeting Participants	List of participants for the appointment, delimited by a semicolon (;).	No	None.
Users	List of users that is received from the list of participants when Oracle CRM On Demand Desktop resolves the primary email of the employees.	No	Oracle CRM On Demand Desktop performs the preliminary resolution when an appointment is saved. To improve the quality of this list, after the next synchronization, the Oracle CRM On Demand server analyzes the processing that occurs on the server.

Oracle CRM On Demand Field	Microsoft Outlook Field	Required	Comments
Contacts	List of contacts that is received from the list of participants that remain after employees are removed from this list when Oracle CRM On Demand Desktop resolves the primary email of the contacts.	No	The logic that applies to the Users field also applies to the Contacts field.
Type	Type that the user specifies when linking the Oracle CRM On Demand record. The default value is Meeting.	No	The user can choose the activity type in the form of the Microsoft Outlook calendar item only for an activity that contains an Activity value of Appointment. The default value is specified in a customization package. You can change this value.
Subject	For more information, see *How Oracle CRM On Demand Desktop Handles an Item That Is Marked Private*.	Yes	The value that is set for a private task is specified in the customization package and is included in the localization resources. For more information on localization resources and customizing localization for Oracle CRM On Demand Desktop, see *Customizing Localization*.
Priority The values are: • 1-High • 2-Medium • 3-Low	Microsoft Outlook priority The values are: • 1-High • 2-Medium • 3-Low	No	For more information, see *How Oracle CRM On Demand Desktop Maps the Priority Field*.
Description	Description A private calendar item does not include a description.	No	None.
Account	Account association that the user specifies when the user links an Oracle CRM On Demand record.	No	None.
Opportunity	Opportunity association that the user specifies when the user links an Oracle CRM On Demand record.	No	None.
StartTime	Start Date/Time	Yes	This is the start time of the Oracle CRM On Demand appointment.
EndTime	End Date/Time	Yes	None.
Duration	Duration	Yes	None.
Meeting Location	Location	No	None.

Oracle CRM On Demand Field	Microsoft Outlook Field	Required	Comments

How Oracle CRM On Demand Desktop Handles an Item That Is Marked Private

Oracle CRM On Demand Desktop handles an item that is marked Private in the following ways:

- **From Microsoft Outlook to Oracle CRM On Demand.**. Assume the user shares a meeting with Oracle CRM On Demand and also adds a check mark to the Private check box to make this meeting private. In this situation, Oracle CRM On Demand Desktop synchronizes the meeting and the setting of the Private check box to the Oracle CRM On Demand server. It performs this synchronization in the same way that it synchronizes any other field. Oracle CRM On Demand does not display the contents of this meeting to any user who interacts with the Oracle CRM On Demand server except the meeting owner.
- **From Oracle CRM On Demand to Microsoft Outlook.** If the user is directly associated with an activity, then a filter on the action object causes Oracle CRM On Demand Desktop to synchronize this activity with Microsoft Outlook even if the Private check box for this field contains a check mark. For example, assume the Private check box in an activity for an account contains a check mark. In this situation, Oracle CRM On Demand Desktop does the following:
  - Downloads and displays this activity for the user who is associated with it.
  - Does not download or display this activity to any user who is not associated with this activity record, even if this user is on the Account Team.

How Oracle CRM On Demand Desktop Maps the Priority Field

The following table describes how Oracle CRM On Demand Desktop maps the Oracle CRM On Demand priority fields with those in Microsoft Outlook.

Oracle CRM On Demand Values	Microsoft Outlook Values
3 - Low	Low
2 - Medium	Normal
1 - High	High

How Oracle CRM On Demand Desktop Maps Fields Between Oracle CRM On Demand Activities and Microsoft Outlook Tasks

This topic describes how Oracle CRM On Demand Desktop maps fields between an Oracle CRM On Demand activity and a Microsoft Outlook task. The following table describes how Oracle CRM On Demand Desktop maps fields between an Oracle CRM On Demand activity and a Microsoft Outlook task.

Oracle CRM On Demand Field	Microsoft Outlook Field	Required	Description
Owner	Owner	Yes	For more information, see *How Oracle CRM On Demand Desktop Maps the Owner Field Between an Oracle CRM On Demand Activity and a Microsoft Outlook Task*.
Type	The type that the user specifies when the user links an Oracle CRM On Demand record. The default value is To Do.	No	When a task is shared, Oracle CRM On Demand Desktop creates an activity in Microsoft Outlook with an Activity value of Task. The default value is specified in the customization package. You can change this value.
Subject	Microsoft Outlook Task Subject	Yes	The value that is set for a private task is specified in the customization package and is included in the localization resources.
Priority The values are: - 1-High - 2-Medium - 3-Low	Microsoft Outlook priority The values are: - 1-High - 2-Medium - 3-Low	No	For more information, see *How Oracle CRM On Demand Desktop Maps the Priority Field*.
Description	Description. A private task does not include a description.	No	If the Private check box is set to allow a shared task, then Oracle CRM On Demand Desktop sets the Comments field to a value that is defined in the customization package, such as Unavailable.
Start Date	Start Date	No	The Start Date for an activity is the start date that the user sets for a task. If the user does not enter the start date, then the start date for the activity is also empty.
CompletedDatetime	Completed Date	No	This is the completed date on the task in Oracle CRM On Demand.
Completed	Completed flag	No	None.
DueDate	Due Date	No	This is the due date on the task in Oracle CRM On Demand.
Status	Status	No	For more information, see *How Oracle CRM On Demand Desktop*

Oracle CRM On Demand Field	Microsoft Outlook Field	Required	Description
			Maps the Status Field of an Activity.
Account	Account association that the user specifies when the user links an Oracle CRM On Demand record.	No	None.
Opportunity	Opportunity association that the user specifies when the user links an Oracle CRM On Demand record.	No	None.
Contacts	Contact association that the user specifies when the user links an Oracle CRM On Demand record.	No	None.

How Oracle CRM On Demand Desktop Maps the Owner Field Between an Oracle CRM On Demand Activity and a Microsoft Outlook Task

This topic describes how Oracle CRM On Demand Desktop maps the owner field between an Oracle CRM On Demand activity and a Microsoft Outlook task. The following logic applies:

- If a shared task is assigned to another employee, then Oracle CRM On Demand Desktop sets the assignee as the owner of the activity.
- If a shared task is assigned to a number of employees, then Oracle CRM On Demand Desktop sets each assignee as the owner of their own activity.
- If a shared task is assigned to a shared contact or to an external person, then Oracle CRM On Demand Desktop sets the employee who created the task as the owner of the activity.
- If a task is created by an external person and assigned to an employee, then when this employee shares the task Oracle CRM On Demand Desktop sets the employee as the owner of the activity.
- If an external person who is not an Oracle employee sends a task to a number of employees who are Oracle CRM On Demand Desktop users, then Oracle CRM On Demand Desktop sets each employee as the owner of the activity and adds the remaining employees to the employee team. Oracle CRM On Demand Desktop performs this work after each employee accepts and shares the task.

How Oracle CRM On Demand Desktop Maps the Status Field of an Activity

The following table describes how Oracle CRM On Demand Desktop maps the status of an Oracle CRM On Demand activity to the status of a Microsoft Outlook task when the task is shared.

Oracle CRM On Demand Status	Microsoft Outlook Status
Not Started	Not Started
In Progress	In Progress
Completed	Completed
Waiting for Someone Else	Waiting on Someone Else
Deferred	Deferred

How Oracle CRM On Demand Desktop Maps the Status Field of an Activity

This topic gives one example of how Oracle CRM On Demand Desktop maps the status field of an activity. Oracle CRM On Demand Desktop performs the following sequence:

1. A user creates a shared task in Microsoft Outlook that contains a status that is In Progress and a Percent Complete that is 0.
2. Oracle CRM On Demand Desktop creates an activity on the Oracle CRM On Demand server that contains a status of In Progress.
3. The user synchronizes with the Oracle CRM On Demand server.
4. On the Oracle CRM On Demand server, Oracle CRM On Demand Desktop changes the status to Requested.
5. The user synchronizes with the Oracle CRM On Demand server again.
6. In Microsoft Outlook, Oracle CRM On Demand Desktop updates the status of the activity to Requested.
7. At this point, the Percent Complete field of the task is 0. Therefore, Oracle CRM On Demand Desktop updates the status of the task to Not Started.

How Oracle CRM On Demand Desktop Maps Fields Between an Oracle CRM On Demand Activity and a Microsoft Outlook Email

The following table describes how Oracle CRM On Demand Desktop maps fields between an Oracle CRM On Demand activity and a Microsoft Outlook email.

Oracle CRM On Demand Field	Required	Microsoft Outlook Field
Type	No	Email
Subject	No	Email Subject
Activity	Yes	Task
Priority	No	Importance.

Oracle CRM On Demand Field	Required	Microsoft Outlook Field
		Oracle CRM On Demand Desktop applies the following mapping: • 1-High in Oracle CRM On Demand data is High in Microsoft Outlook • 2-Medium in Oracle CRM On Demand data is Normal in Microsoft Outlook • 3-Low in Oracle CRM On Demand data is Low in Microsoft Outlook For more information, see *How Oracle CRM On Demand Desktop Maps the Priority Field*.
Account Id	No	The Primary account association that the user specifies when the user creates the activity.
Opportunity Id	No	The opportunity association that the user specifies when the user creates the activity.
Attachment	No	The original email that Oracle CRM On Demand Desktop saves as an attachment to the activity.

12 XML Files Reference

XML Files Reference

This chapter describes the code in the XML files that Oracle CRM On Demand Desktop includes in the customization package. It includes the following topics:

- *XML Code to Map a Field in Microsoft Outlook Storage*
- *XML Code to Map a Field in DB_FACADE storage*
- *XML Code to Customize the Data That Oracle CRM On Demand Desktop Deletes If You Remove Oracle CRM On Demand Desktop*
- *XML Code to Customize Synchronization*
- *XML Code to Customize Forms*
- *XML Code to Customize Toolbars*
- *XML Code to Customize Dialog Boxes*
- *XML Code to Customize Views Used for Microsoft Outlook Explorer List*
- *XML Code to Customize Views Used for Objects Forms*
- *XML Code to Customize the Sales Book Control*
- *XML Code for Meta Information*

For more information on customization, see *Customizing Oracle CRM On Demand Desktop*.

XML Code to Map a Field in Microsoft Outlook Storage

This topic describes the code of the od_basic_mapping.xml file. It includes the following information:

- *Example Code of the od_basic_mapping.xml File*
- *Type Element of the od_basic_mapping.xml File*
- *Form Element of the od_basic_mapping.xml File*
- *Alt Message Classes Element of the od_basic_mapping.xml File*
- *Custom Views Element of the od_basic_mapping.xml File*
- *Field Element of the od_basic_mapping.xml File*
- *Writer Element of the od_basic_mapping.xml File*

For more information, see *Customizing How Oracle CRM On Demand Desktop Maps Fields*.

Example Code of the od_basic_mapping.xml File

This topic gives one example of code that Oracle CRM On Demand Desktop uses in the od_basic_mapping.xml file. You might use this feature differently, depending on your business model. The following code is an example of the od_basic_mapping.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<sd2_meta>
 <types>
 <type id="Contact" predefined_folder="10">
 <form message_class="IPM.Contact.OnDemand.Contact"
icon="type_image:Contact:16" large_icon="type_image:Contact:32"
display_name="Contact">OnDemand Contact</form>
 <alt_messageclasses>
 <alt_messageclass ext="Private" display_name="Private Contact"
icon="type_image:Contact.Private:16"
large_icon="type_image:Contact.Private:32">OnDemand Contact</alt_messageclass>
 </alt_messageclasses>
 <custom_views default_name="#view_crm_and_personal_contacts">
 <view id="all_contacts" name="#view_crm_and_personal_contacts"></view>
 </custom_views>
 <field id=" First Name ">
 <reader class="mapi_std">
 <mapi_tag id="0x3A060000"></mapi_tag>
 <convertor class="string"></convertor>
 </reader>
 <writer class="Microsoft Outlook_std">
 <Microsoft Outlook_field id=" FirstName "></Microsoft Outlook_field>
 <convertor class="string"></convertor>
 </writer>
 </field>
 <field id="Location">
 <reader class="mapi_user">
 <user_field id="od Location" ol_field_type="1"></user_field>
 <convertor class="string"></convertor>
 </reader>
 <writer class="Microsoft Outlook_user">
 <user_field id="od Location" ol_field_type="1"></user_field>
 <convertor class="string"></convertor>
 </writer>
 </field>
 </type>
 </types>
</sd2_meta>
```

Type Element of the od_basic_mapping.xml File

The type element defines the Oracle CRM On Demand object that Oracle CRM On Demand Desktop maps to Microsoft Outlook. Oracle CRM On Demand Desktop includes the following attributes in the type element:

- **id.** Defines the ID or name of the Oracle CRM On Demand object that Oracle CRM On Demand Desktop maps to Microsoft Outlook.
- **display_name.** Defines the name of the folder that Oracle CRM On Demand Desktop displays in the Microsoft Outlook tree view that Oracle CRM On Demand Desktop uses to store the records of the Oracle CRM On Demand object.
- **folder_type.** Defines the type of the Microsoft Outlook folder. The value for the folder_type attribute is typically 10 for a custom Oracle CRM On Demand object.
- **hidden_folder.** Determines if the folder that the display_name attribute specifies is visible in the Microsoft Outlook tree view.
- **predefined_folder.** Defines the type of Microsoft Outlook folder. If you must store Oracle CRM On Demand objects in a native Microsoft Outlook folder, then Oracle CRM On Demand Desktop uses the predefined_folder attribute.

- **prohibit_user_modification.** Prohibits the modification to an object that is declared in the type element. The default value is False.
- **ver.** Used during development. Oracle CRM On Demand Desktop does not apply any change to the description for the object that the ID attribute defines until the value of the ver attribute is increased.

Form Element of the od_basic_mapping.xml File

The form element defines the ID of the form that Oracle CRM On Demand Desktop uses to display the object that is defined in the type element. The form with the corresponding ID is described in the forms_xx.xml file. For more information, see *XML Code to Customize Forms*.

Oracle CRM On Demand Desktop includes the following attributes in the form element:

- **message_class.** Defines the Microsoft Outlook message class for the form. The message class is an extension of native Microsoft Outlook message classes, for example, IPM.Contact.OnDemand.Contact, or IPM.Contact.OnDemand.Account.
- **icon.** Defines the icon that Oracle CRM On Demand Desktop uses to display the following objects:
  - The object that is defined in the type element in a Microsoft Outlook table view.
  - The icon views for small icons and list view modes.
  - A form caption icon, which displays in the top-left corner in front of the form caption. Oracle CRM On Demand Desktop stores icons in the platform_images.xml file. The value of the icon attribute must be the key value of the required image.
- **large_icon.** Defines the icon to use to represent the object that is defined in the type element when the user uses the large icons mode. Oracle CRM On Demand Desktop stores icons in the platform_images.xml file. The value of the large_icon attribute must be the key value of the necessary image.
- **display_name.** Defines the name that Oracle CRM On Demand Desktop displays on the form caption.

Alt Message Classes Element of the od_basic_mapping.xml File

To define an alternative message class to an object in Microsoft Outlook, you can use the alt_messageclasses element. This element is useful if an Oracle CRM On Demand object might be in one of several different states. For example, a contact might be shared or not shared.

This element must contain a set of alt_messageclass tags that define each state of an object. The alt_messageclass element can also define a form that you use for an object with this message class. In this way, you can use different forms for the same object but in different states. Each alt_messageclass element includes the following attributes:

- **Ext.** Indicates an extension that is added to an original message class.
- **display_name.** The same as the message_class element.
- **icon.** The same as the message_class element.
- **large_icon.** The same as the message_class element.

Custom Views Element of the od_basic_mapping.xml File

The custom_views element defines a set of custom Microsoft Outlook views that Oracle CRM On Demand Desktop applies for the object that is defined in the type element. The default_name attribute sets the default view that Oracle CRM On Demand Desktop applies after the installation.

Oracle CRM On Demand Desktop includes the following tags in the custom_views element:

- **view.** Describes each view. Each view element includes the following attributes:
  - **id.** Defines the ID of the view. The view is described in the views.xml file.
  - **name.** Defines the name of the view that is specified in the id attribute to display in the Current view menu. To access this menu in Microsoft Outlook, the user chooses the View menu, Arrange By, and then the Current view menu.

  The following code is an example of the custom_views element:

```
<custom_views default_name="All Activities">
 <view id="all_activities" name="All Activities"></view>
 <view id="all_activities_by_duedate" name="Activities by Due Date"></view>
 <view id="all_activities_by_owner" name=" Activities by Owner"></view>
 <view id="all_activities_by_priority" name="Activities by Priority"></view>
</custom_views>
```

Field Element of the od_basic_mapping.xml File

The field element describes the field mapping. Because the field element describes one field, the number of field tags must be the same as the number of fields that are mapped. The following attributes are included in the field element:

- **id.** Defines the field identifier, for example, the API name of the field. To assign a control to this field on a form, Oracle CRM On Demand Desktop also uses the value of this attribute in the forms_xx.xml file.
- **ver.** Used during development. Oracle CRM On Demand Desktop does not apply any change to the field description until the value of the ver attribute is increased.

Writer Element of the od_basic_mapping.xml File

The writer element defines write access to the field that Oracle CRM On Demand Desktop maps to Microsoft Outlook. It contains only the class attribute. The following values are available for the class attribute:

- binhex_link
- custom:links_first
- multiwriter
- Microsoft Outlook_document_content
- Microsoft Outlook_document_filename
- Microsoft Outlook_recipients
- Microsoft Outlook_std

- Microsoft Outlook_user

The writer element can contain the following tags:

- link_writer
- resolved_writer
- Microsoft Outlook_field
- user_field
- convertor
- writer

XML Code to Map a Field in DB_FACADE storage

For objects that use DB FACADE storage, the mapping of a field is different and has the following structure:

```
<database>
 <types>
 <type id="Currency" icon="type_image:Currency:16">
 <field id="Name">
 <type>
 <simple type="string" />
 </type>
 </field>
 </type>
 </types>
</database>
```

The following attributes must be defined for the <type> element:

- **id.** Defines the ID or name of the Oracle CRM On Demand object that Oracle CRM On Demand Desktop maps to when DB FACADE storage is used.
- **icon.** Defines the image used to accompany an object's records in a list. The id attribute for the <field> element defines the ID or name of the Oracle CRM On Demand object's field.

The type attribute for <simple> element defines the type of the Oracle CRM On Demand object's field. The type attribute can have the following values for corresponding data types, for example, string for a string data type:

- string
- integer
- double
- boolean
- datetime

For the ID fields, that is, the fields used in links, the following construction exists, and you must use this construction when a field contains a link to the object placed in DB FACADE storage.

```
<field id="Primary Employee Id">
 <type>
 <foreign_key>
 <type_id>Employee</type_id>
 </foreign_key>
 </type>
</field>
```

However, when a field contains a link to the object placed in Outlook storage, use the following construction:

```
<field id="Primary Contact Id">
 <type>
 <simple type="binary" />
 </type>
</field>
```

XML Code to Customize the Data That Oracle CRM On Demand Desktop Deletes If You Remove Oracle CRM On Demand Desktop

This topic gives one example of code that Oracle CRM On Demand Desktop uses in the platform_configuration.xml file. You might use this feature differently, depending on your business model. For more information, see *Specifying Which Data Oracle CRM On Demand Desktop Removes from Microsoft Outlook*.

The following code is an example of the platform_configuration.xml file:

```
<platform>
<items_remover>
<rules>
 <type id="Mail" rule="skip" />
 <type id="Task" rule="skip" />
 <type id="Event" rule="skip" />
 <type id="Action" rule="script" language="JScript">
 <![CDATA[

 //"Calendar and Activities";
 //"To Do and Activities";
 //"Activities Only";
 var pim_id = item.PIMObjectId;
 if (!item["Appt PIM Flag"] && pim_id != null)
 {
 var pim_item = open_item(pim_id);
 if (pim_item != null)
 pim_item.remove();
 }

 true; // allow to process this item

 ]]>
 </type>
</rules>
</items_remover>
</platform>
```

XML Code to Customize Synchronization

This topic describes the code of the connector_configuration.xml file. It includes the following information:

- *Example Code for the connector_configuration.xml File*
- *Types Element of the connector_configuration.xml File*

- *Type Element of the connector_configuration.xml File*
- *View Element of the connector_configuration.xml File*
- *Synchronizer Element of the connector_configuration.xml File*
- *Links Element of the connector_configuration.xml File*
- *Natural Key Element of the connector_configuration.xml File*
- *Filter Presets Element of the connector_configuration.xml File*

For more information, see *Customizing Synchronization*.

Example Code for the connector_configuration.xml File

This topic gives one example of code that Oracle CRM On Demand Desktop uses in the connector_configuration.xml file. You might use this feature differently, depending on your business model. The following code is an example of the connector_configuration.xml file:

```
<root>
 <types>
 <type id="Opportunity">
 <view label="Opportunity" label_plural="Opportunities"
small_icon="type_image:Opportunity:16" normal_icon="type_image:Opportunity:24"
large_icon="type_image:Opportunity:48"></view>
 <synchronizer name_format=":[:(Name):]">
 <links>
 <link>Account Id</link>
 <link>Currency Code</link>
 </links>
 <natural_keys>
 <natural_key>
 <field>Name</field>
 </natural_key>
 </natural_keys>
 </synchronizer>
 </type>
 </types>
 <filter_presets>
 <preset name="Test filters">
 <type id="Action">
 <group link="and">
 <binary field="Planned" condition="ge">
 <value type="function">today</value>
 </binary>
 </group>
 </type>
 </preset>
 </filter_presets>
</root>
```

Types Element of the connector_configuration.xml File

The types element describes the types of objects to synchronize. It does not contain attributes. It does contain a set of type tags. You must describe these types in the od_basic_mapping.xml file.

Type Element of the connector_configuration.xml File

The type element describes the type to synchronize. You must describe it in the od_basic_mapping.xml file. It includes the id attribute, which defines the ID of the object.

The type element must contain the following tags:

- view
- synchronizer

View Element of the connector_configuration.xml File

The view element defines the type in the user interface, in particular, the Filter Records tab on the Synchronization Control Panel dialog box.

The view element includes the following attributes:

- **label.** The label that Oracle CRM On Demand Desktop uses to display this object in the Synchronization Control Panel dialog box if Oracle CRM On Demand Desktop cannot resolve the name of this object.
- **label_plural.** The label that Oracle CRM On Demand Desktop uses to display this object in the Synchronization Control Panel dialog box if the label is most appropriately displayed in the plural form.
- **small icon.** Defines a 16-by-16 pixel icon that Oracle CRM On Demand Desktop uses for this object on the Synchronization Control Panel dialog box.
- **normal_icon.** Defines the icon that Oracle CRM On Demand Desktop displays next to the object type in the Filter Records tab of the Synchronization Control Panel dialog box.
- **large_icon.** Defines the icon that Oracle CRM On Demand Desktop displays on the Oracle CRM On Demand Desktop Synchronization dialog box while Oracle CRM On Demand Desktop synchronizes this type of object.
- **suppress_sync_ui.** If suppress_sync_ui is true, then this attribute hides the object of this type from the Filter Records tab of the Synchronization Control Panel dialog box. If suppress_sync_ui is not defined, then Oracle CRM On Demand Desktop applies the false value by default.

For more information, see *Controlling the Object Types That Oracle CRM On Demand Desktop Displays inthe Filter Records Tab*.

Synchronizer Element of the connector_configuration.xml File

The synchronizer element describes more settings for synchronization. It defines attributes that the synchronization engine requires. For example, relations between objects or criteria to identify duplicate objects.

The synchronizer element includes the name_format attribute, which defines the format of the output string for objects of this type. Oracle CRM On Demand Desktop uses this string if objects of this type are displayed on the Check Issues, Resolve Conflicts, Resolve Duplicates, or Confirm Synchronization tab of the Control Panel in the Oracle CRM On Demand Desktop application.

The synchronizer element can contain the following tags:

- links

- natural_keys

Links Element of the connector_configuration.xml File

The links element describes the references between types. You must define it in the synchronizer element. Note the following requirements:

- You must use the links element to describe all fields that Oracle CRM On Demand Desktop uses to store references between objects. A link field is an example of a field that Oracle CRM On Demand Desktop uses to store a reference between objects.
- You must describe all links in the links element.

The links and link tags do not contain attributes.

Example Code of the Links Element

The following code is an example of the links element:

```
<links>
 <link>Account Id</link>
 <link>Opportunity Id</link>
 <link>Created By</link>
 <link>Primary Owner Id</link>
</links>
```

Natural Key Element of the connector_configuration.xml File

The natural_key element is defined in the synchronizer element. You use it to configure criteria to identify duplicated records during synchronization. The natural_key element contains the following:

- A set of natural_key tags that describes the criteria. Oracle CRM On Demand Desktop uses OR logic for all criteria that the natural_key element describes.
- A set of field tags, each of which contains a field name that Oracle CRM On Demand Desktop examines to identify duplicates. Oracle CRM On Demand Desktop uses AND logic for all field tags.

Example Code of the Natural Key Element

The following code is an example of the natural_keys element:

```
<natural_keys>
 <natural_key>
 <field>First Name</field>
 <field>Last Name</field>
 </natural_key>
 <natural_key>
 <field>Email Address</field>
 </natural_key>
</natural_keys>
```

In this code, two objects are detected as duplicates if one of the following situations is true:

- First Name AND Last Name contain the same values
- Email Address fields contain the same values

Filter Presets Element of the connector_configuration.xml File

The filter_presets element contains predefined filter criteria. The preset element describes these criteria. The preset element includes a single attribute, name, which defines the name for these criteria. The preset element contains a set of type tags that define filter criteria for each type.

The type element defines the object type to which Oracle CRM On Demand Desktop applies this filter criteria. You must define the object type in the id attribute of this element. The group element describes a group of criteria.

Example Code of the Filter Presets Element

The following code is an example of the usage of the filter_presets element of the connector_configuration.xml file:

```
<filter_presets>
 <preset name="Test filters">
 <type id="Opportunity">
 <group link="and">
 <binary field="Status" condition="in">
 <value type="array">
 <value type="string">Accepted</value>
 <value type="string">Pending</value>
 <value type="string">Rejected</value>
 <value type="string">Rerouted</value>
 </value>
 </binary>
 </group>
 </type>
 <type id="Action">
 <group link="and">
 <binary field="Planned" condition="ge">
 <value type="function">today</value>
 </binary>
 </group>
 </type>
 </preset>
</filter_presets>
```

Example Code to Set the Size and Type of Field

This topic describes code you can use to set the size and type of field. The following code is an example usage of the group element of the connector_configuration.xml file to set the size and type of field:

```
<group link="and">
 <binary field="FileSize" condition="le">
 <value type="integer">5242880</value>
 </binary>
 <group link="or">
 <binary field="FileExt" condition="eq">
 <value type="string">doc</value>
 </binary>
 <binary field="FileExt" condition="eq">
 <value type="string">docx</value>
 </binary>
 <binary field="FileExt" condition="eq">
 <value type="string">xls</value>
 </binary>
 <binary field="FileExt" condition="eq">
 <value type="string">xlsx</value>
 </binary>
 <binary field="FileExt" condition="eq">
```

```
     <value type="string">msg</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">txt</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">rtf</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">html</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">ppt</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">pptx</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">pdf</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">mht</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">mpp</value>
     </binary>
     <binary field="FileExt" condition="eq">
     <value type="string">vsd</value>
     </binary>
     </group>
    </group>
```

XML Code to Customize Forms

This topic describes the code of the forms_12.xml files. It includes the following information:

- *Form Element of the forms_xx.xml File*
- *Script Element of the forms_xx.xml File*
- *Info Bar Element of the forms_xx.xml File*
- *Page Element of the forms_xx.xml File*
- *Stack Element of the forms_xx.xml File*
- *Control Tags of the forms_xx.xml File*
- *Types of Controls for the Control Element of the forms_xx.xml File*

For more information, see *Customizing a Form*.

Form Element of the forms_xx.xml File

The form element describes the user interface for each custom Microsoft Outlook form. It can include the following attribute: id. It defines the unique name for the current form.

Standard code is present in the <script> element, which defines the form handler. An example of this is "contact_note() in the provided code. The form handler must be declared in forms.js file.

The form element contains the following tags:

- script
- info_bar
- page

Each of these tags describe a specific part of the form. You can ignore each element.

Example Code of the Form Element

The following code is an example of the form element.

This code defines the Contact Note form:

```
<forms>
 <form id="OD Contact Note" >
 <script>
 <![CDATA[
 include("forms.js", "forms");
 var ctx = {
 "application": application,
 "ui": ui,
 "application_script": application_script,
 "form": form
 };
 var current_form = new forms.contact_note(forms.create_form_ctx(ctx));
 ]]>
 </script>
 <page id="General" tag="0x10A6" min_height="155" min_width="520">
 <cell>
 <stack layout="vert" padding="5">
 <cell>
 <stack layout="horz" spacing="3">
 <cell size="65">
 <stack layout="vert" spacing="5">
 <cell size="22">
 <static id="0x20004" tab_order="2">
 <text>#lbl_type</text>
 </static>
 </cell>
 <cell size="22">
 <static id="0x20004" tab_order="4">
 <text>#lbl_description</text>
 </static>
 </cell>
 </stack>
 </cell>
 <cell>
 <stack layout="vert" spacing="5">
 <cell size="22">
 <combobox id="NoteType" tab_order="3">
 <field>Note Type</field>
 <source type="Contact.Contact_NoteNote
TypePicklist" field="Value" format=":[:(Label):]"></source>
 </combobox>
 </cell>
 <cell>
 <control id="0x103f" tab_order="2"></control>
 </cell>
 </stack>
 </cell>
 </stack>
 </cell>
 </stack>
```

```
  </cell>
  </page>
  </form>
</forms>
```

Script Element of the forms_xx.xml File

The script element stores all JavaScript functions that Oracle CRM On Demand Desktop uses in the user interface of the current form. You can use these scripts for different purposes, as required. You must describe all script functions in the CDATA section of the validation_rules element.

Example Code of the Script Element

The following code is an example of the script element:

```
<script>
<![CDATA[
 include("forms.js", "forms");
 var ctx = {
 "application": application,
 "ui": ui,
 "application_script": application_script,
 "form": form
 };
 var current_form = new forms.contact_note(forms.create_form_ctx(ctx));
]]>
</script>
```

Info Bar Element of the forms_xx.xml File

The info_bar element is an alternative to the page element. You can use it to add a layout that the original form layout does not determine. You can use the info_bar element to extend the original form but not to modify the original form. For example, the predefined mail form includes an Info Bar. The Info Bar element contains the Share With On Demand content that Oracle CRM On Demand Desktop displays if the user clicks the Share Bar. The Info Bar does not affect other parts of the form.

Note the following differences between how you can use the info_bar element and the page element:

- To customize only a section of a predefined Microsoft Outlook form, you can use the info_bar tab, for example, on the Mail, Task, or Calendar, forms.
- To customize an entire form, you can use the page element. For example, the page element replaces the native Contact form in Microsoft Outlook with a custom Oracle CRM On Demand Desktop form.

Page Element of the forms_xx.xml File

The page element describes the layout of the form. The layout is defined by a set of cell tags that contain the user interface elements and data that make up the form. Note the following requirements for the cell element:

- A cell element can be empty or can contain a stack of cell elements in the stack element or a control.

- A cell element can contain a user interface element or a piece of data from a native Microsoft Outlook object and from a standard or custom Oracle CRM On Demand object. If you must place more than one object in a cell, then you must use a stack element in the cell element.

A page element can contain only one cell. The page element can include the following attributes:

- **id.** Contains the page name and is used nowhere else.
- **tag.** Contains the control ID of the first control on the page of a standard form. It is a page that Oracle CRM On Demand Desktop uses to apply a customized layout. For example, the first control on a General tab of a Contact form uses id=0x402, Details page - 0?11cf, and so on.
- **min_height.** Describes the minimum height, in pixels.
- **min_width.** Describes the minimum width, in pixels.

Example Code of the Page Element

The following code is an example of the page element:

```
<page id="General" tag="0x0402">
 <cell>
 </cell>
</page>
```

Cell Element of the Page Element of the forms_xx.xml File

The cell element is a layout cell that contains a stack of cells or a control. It defines the position of the stack. It includes the following attributes:

- **size.** Defines the cell size, in pixels. You must define the size if the cell is situated in a stack of cells:
  - In a stack of cells that is arranged horizontally, the size defines the cell width.
  - In a stack of cells that is arranged vertically, the size defines the height.
- **attraction.** Defines cell docking. You must define the attraction if the cell is in a stack of cells. The following values are available:
  - **near.** The cell docks to the left side of a horizontal stack of objects or to the top of a vertical stack of objects.
  - **far.** The cell docks to the right side of a horizontal stack of objects or to the bottom of a vertical stack of objects.
  - **both.** The cell consumes the entire free space. If more than one cell is set to both, then Oracle CRM On Demand Desktop divides the free space equally between all cells that are set to both.

The default value is near if the cell size is defined. The default value is both if there is no size.

Stack Element of the forms_xx.xml File

The stack element defines a stack of cells that is placed in a cell. Oracle CRM On Demand Desktop docks a cell that contains the near attribute in the following ways:

- To the top border in a vertical stack
- To the left border in a horizontal stack

The stack element includes the following attributes:

- **layout.** Defines the type of stack. The possible values include horz (horizontal) or vert (vertical).

- **spacing.** Defines the space between cells in a stack. The default value is 0.
- **padding.** Defines the space between the cell and the stack border. The cell defines the stack border. This cell contains the stack. The default value is 0.

Example Code of the Stack Element

The following code is an example of the stack element:

```
<page id="General" tag="0x0402">
 <cell>
 <stack layout="horz">
 <cell size="25"/> <!--attraction="near"-->
 <cell size="30"/>
 <cell/> <!--attraction="both"-->
 <cell attraction="both">
 <cell size="20" attraction="far">
 </stack>
 </cell>
</page>
```

Control Tags of the forms_xx.xml File

The control element defines a control that is located in a cell. It includes the following attributes:

- **id.** Defines the control ID. If you use the id attribute to specify the ID of a native Microsoft Outlook control, then Oracle CRM On Demand Desktop requires no more attributes except tab_order, if necessary. The class attribute defines the control type.
- **tab_order.** Defines the tab order of the control. Oracle CRM On Demand Desktop displays the control that contains the smallest tab_order value as the first control in the tab order. You can start at 1. If the control contains no tab_order, then Oracle CRM On Demand Desktop does not include the control in the tab order.
- **allow_negative.** The multi_currency control is the only control used with this attribute.
- **caption.** For list control only.
- **type.** The outlook_view control is the only control used with this attribute.
- **view_id.** The outlook_view control is the only control used with this attribute.

The following controls support the control element:

- button
- check box
- gradient_checkbox

Control Element Tags

The control element includes the following tags. These tags depend on the name of the parent control element:

- **source.** Used for multivalue controls, such as combobox, lookup, and mvg_primary_selector. It defines the objects that Oracle CRM On Demand Desktop displays in this control. It includes the following attributes:
    - **type.** Defines the type of an object that Oracle CRM On Demand Desktop displays in the current control. Objects of this type must be described in the basic_mapping.xml file.
    - **display_format.** Defines what object fields Oracle CRM On Demand Desktop displays in a control. Used for the mvg_primary_selector control only.
    - **format.** The same as the display_format element but it is applied to other controls.

  - **field.** Defines what object field is chosen as a value in a box. Used for the combobox control only.
  - **left_id.** Defines a field on a related object where Oracle CRM On Demand Desktop stores the ID of the parent object. This is the ID of the current object on which the left_id control is used. Used for the mvg_primary_selector control only.
  - **item_value.** Defines a field on a related object where Oracle CRM On Demand Desktop stores the ID of the parent object. This is the ID of the chosen object. Used for the mvg_primary_selector control only.
- **text.** Defines the text that Oracle CRM On Demand Desktop uses for the control. You use this element primarily for the label control.
- **field.** Contains the field identifier with which this control works:
  - This field element must be described in the od_basic_mapping.xml file.
  - Contains one attribute, value, which Oracle CRM On Demand Desktop uses for edit controls only. This attribute describes the type of the value for the field.
  - The following values are available: string, binary, int, double, or currency.
  - The API name field is an example of a field identifier.

Example Code of the Control Element

The following code is an example of the edit control element:

```
<cell>
 <edit id="0x10000" max_chars="100" tab_order="7">
 <field value="string">Name</field>
 </edit>
</cell>
```

Types of Controls for the Control Element of the forms_xx.xml File

This topic describes the types of controls you can configure for the control element of the forms_xx.xml file. It includes the following information:

- *Combobox Control of the forms_xx.xml File*
- *Dropdown Control of the forms_xx.xml File*
- *multi_currency Control of the forms_xx.xml File*
- *MVG Primary Selector Control of the forms_xx.xml File*
- *outlook_view Control of the forms_xx.xml File*
- *data_view Control of the forms_xx.xml File*
- *Subform Control of the forms_xx.xml File*
- *Web_page Control of the forms_xx.xml File*

Names of the Control Element of the forms_xx.xml File

The following table describes the names of the control element of the forms_12.xml file.

Value	Description
button	A button control. If you use this control, then it is not necessary to use a field element.

Value	Description
checkbox	A check box control. The field that the field element assigns to this control must include the following configuration: • The name of the field • The field type must be Boolean
combobox	A simple control that allows the user to choose any value from a list. For more information, see *Combobox Control of the forms_xx.xml File*.
datetime	A datetime control that allows the user to choose a date from a calendar. The field element must contain the date or datetime field.
dropdown	A control that displays a menu when the user clicks the control. For example, if the user clicks Addresses on the contact form, then Oracle CRM On Demand Desktop displays Personal Addresses or Business Addresses. For more information, see *Dropdown Control of the forms_xx.xml File*.
edge	A panel control that includes a border. If you use this control in a cell where the size is 1, such as with a separator, then it is not necessary to use a field element.
edit	A simple edit box control.
gradient_checkbox	A control that behaves like a check box control, but uses a different graphical interface. This control is displayed on the Sharing bar.
autocomplete	A control that the user can use to choose any object. Oracle CRM On Demand Desktop uses an autocomplete control to establish a relationship between objects, for example, to link an account with a contact.
multi_currency	A control that Oracle CRM On Demand Desktop uses to display the values from more than one field, such as price, revenue, and so on. An example usage of the multi_currency control is where Oracle CRM On Demand Desktop must display the amount and currency values in a single field. For more information, see *multi_currency Control of the forms_xx.xml File*.
mvg_primary_selector	A control that Oracle CRM On Demand Desktop uses to display the Primary association in a many-to-many relationship. For more information, see *MVG Primary Selector Control of the forms_xx.xml File*.
outlook_view	A control that Oracle CRM On Demand Desktop uses to display a set of related items in the Microsoft Outlook view or form.
static	A static control that you can use as a label on a form. If you use this control, then it is not necessary to use a field element.
subform	A group of controls that you can use to display the fields of one object on the form of another object. For example, you can display an Oracle CRM On Demand activity on the native form for a Microsoft Outlook or calendar item. For more information, see *Subform Control of the forms_xx.xml File*.

Combobox Control of the forms_xx.xml File

If you set the name of the control element to combobox, then you must define the following tags in the control element:

- **source.** Describes the list values for this control. The source element includes the following attributes:
  - **type.** Contains the ID of the list that the od_basic_mapping.xml file describes.
  - **field.** Contains the name of the field that Oracle CRM On Demand Desktop displays as a list value.
  - **format.** Defines the mask for this field output. Attributes are usually the same for all lists. It has the following format:

    ```
    field="Value"

    format=":[:(Label):]"
    ```

    Although Label is a variable, you must specify it as an absolute value.
- **field.** A field of an object that stores a value that the user chooses in a list.

Example Code of the Combobox Control

The following code illustrates the use of the combobox control:

```
<combobox id="0x20105">
 <source type="ContactLeadSourcePicklist" field="Value" format=":[:(Label):]"/>
 <field>LeadSource</field>
</combobox>
```

Dropdown Control of the forms_xx.xml File

The dropdown control of the forms_xx.xml file is a button that includes menu options. If the user clicks this button, then Oracle CRM On Demand Desktop displays the menu. Menu items for this menu are added to scripts.

If you set the name of the control element to dropdown, then you must define the following tags in the control element:

- **control.** Contains more tags that you can use to describe the dropdown control.
- **text.** The value in the control element that you can use to define the text of the dropdown control.

You can use the caption attribute to define the caption for the dropdown control.

Example Code of the Dropdown Control

The following code illustrates the use of the dropdown control:

```
<cell size="22">
 <dropdown id="dd_contacts" caption="#lbl_contacts" tab_order="1"
visible="false"></dropdown>

</cell>
```

Autocomplete Control of the forms_xx.xml File

If you set the name of the control element to autocomplete, then you must define the following tags in the control element:

- **source.** Describes the list values for this Autocomplete control. The source element includes the following attributes:
  - **type.** Contains the ID of an object that Oracle CRM On Demand Desktop uses in this control.
  - **format.** Defines the mask for this field output, for example:

```
format=":[:(First Name) :]:[:(Last Name):]"
```

- **field.** The field of an object that stores the ID of the object that the user chooses in a lookup object.

Example Code of the Autocomplete Control

The following code illustrates the use of the Autocomplete control:

```
<autocomplete id="AccountId" tab_order="57">
 <field>AccountId</field>
 <source type="Account" format=":[:(AccountName):]"></source>
</autocomplete>
```

multi_currency Control of the forms_xx.xml File

If you set the name of the control element to multi_currency, then you must define the following tags in the control element:

- **value_field.** Contains the amount field name.
- **currency_field.** Contains the currency field name.
- **exchangedate_field.** Contains the exchange date field value. This element is optional.

Example Code of the multi_currency Control

The following code illustrates the use of the multi_currency control:

```
<multi_currency id="AnnualRevenues" tab_order="28">
 <currency_field>CurrencyCode</currency_field>
 <value_field>AnnualRevenues</value_field>
</multi_currency>
```

MVG Primary Selector Control of the forms_xx.xml File

If you set the class attribute of the control element to mvg_primary_selector, then you must define the following tags in the control element:

- **source.** The behavior is similar to the lookup control. The source element includes the following attributes:
    - **type.** Defines the many-to-many association ID that Oracle CRM On Demand Desktop uses for this control.
    - **linking_field.** Contains the field name of this association where the ID of the parent object is saved.
    - **flag_field.** Contains the field name that Oracle CRM On Demand Desktop uses to set the primary flag.
    - **display_format.** This attribute defines how to display linked objects.

      The objects of the fields are First Name and Last Name, and these objects are linked to the item_value attribute. The values in this attribute take the following format:

      ```
      ":[:(First Name) :]:[:(Last Name):]"
      ```

outlook_view Control of the forms_xx.xml File

If you set the name of the control element to outlook_view, then you must define the following attributes in the control element:

- **view_id.** Contains the ID of the view that Oracle CRM On Demand Desktop uses for this control. The views.xml file describes the view IDs.

- **type.** Defines the ID of the type of objects to display in this control. The od_basic_mapping.xml file describes these objects.

If you set the name of the control element to outlook_view, then you must define the following element in the control element:

- **dasl.** Contains filters that Oracle CRM On Demand Desktop applies to this view. This filter must be in dasl format. For more information about dasl, see the relevant Microsoft documentation. Consider the following example:

```
<outlook_view id="contacts_view" view_id="contacts:form_view"
type="Contact.Account.Association" tab_order="41">

 <dasl><![CDATA[("http://schemas.microsoft.com/mapi/string/
{00020329-0000-0000-C000-000000000046}/od%20AccountId" = '{{id|s}}') AND "http:/
/schemas.microsoft.com/mapi/string/{00020329-0000-0000-C000-000000000046}/
od%20RightStatus" &lt;&gt; 'deleted' AND "http://schemas.microsoft.com/mapi/
string/{00020329-0000-0000-C000-000000000046}/od%20LeftStatus" &lt;&gt;
'unsaved']]></dasl>
 </outlook_view>
```

data_view Control of the forms_xx.xml File

Data_view control has the following structure:

```
<data_view id="activities_view" tab_order="69">
 <source type="auto" name="ActionJointContact" />
 <view id="activities:form_view-mvg" />
 <restriction>
 <group link="and">
 <binary field="ContactId" condition="eq">
 <value type="variable">id()</value>
 </binary>
 <binary field="ContactStatus" condition="ne">
 <value type="string">unsaved</value>
 </binary>
 <binary field="ActionAssocStatus" condition="ne">
 <value type="string">deleted</value>
 </binary>
 </group>
 </restriction>
</data_view>
```

If you set the name of the control element to data_view, then you must define the following attributes in the data_view element:

- **id.** Contains the ID of the view that Oracle CRM On Demand Desktop uses for this control. The views.xml file describes the view IDs.
- **tab_order.** Defines the tab order of the control

You must define the following attributes in the source element:

- **type.** Specifies the storage of the object for the view. It can be outlook, database or auto.
- **name.** Specifies the name of the datasource or object from On Demand basic mapping.

You must define the following attribute in the view element:

- id. Indicates the name of the view representation defined in views.xml.

The restriction block imposes filtering rules on the data set that is passed to data_view control.

The group element link attribute specifies whether the conditions should be applied as a whole or separately.

You must define the following attributes in the binary element:

- **field.** Indicates the field you want to filter by.
- **condition.** Indicates the compare operator that you want to use. Possible options are:
  - lt
  - gt
  - eq
  - ne

  You specify the type of the field in the value element type attribute.

  The data_view control might use the datasource name instead of the object name in the name attribute of the <source> element. The datasource is required if the right type object in the association is defined in outlook storage. For example, in the association Opportunity.Contact.Association, the right object is Contact, and it is defined in Outlook storage so the datasource is needed. Datasources are defined in data_source.xml file, in the joint element, with the following format:

```
<data_source name="ActionJointContact" type_id="Action.Contact.Association"
resolve_rate="1"/>
```

  where:

  - **name.** Indicates the name for the datasource to be used in data_view control.
  - **type_id.** Indicates the association name from the basic_mapping.xml file.

Subform Control of the forms_xx.xml File

The following code illustrates the use of the subform control:

```
<cell size="22">
 <subform id="activity_subform">
 <cell size="22">
 <stack layout="horz" spacing="20" padding="6">
 <cell>
 <static id="ActivityLabel">
 <text>Activity Name:</text>
 </static>
 </cell>
 <cell>
 <edit id="ActivityName">
 <field>Name</field>
 </edit>
 </cell>
 </stack>
 </cell>
 </control>
</cell>
```

Web_page Control of the forms_xx.xml File

If you set the name of the control element to web_page, then you must define the url attribute in the control element. The following code illustrates the use of the web_page control:

```
<control class="web_page" id="linkedin_search">
 <url>http://www.linkedin.com/</url>
```

```
</control>
```

You can define a static or a dynamic URL as the value of the url attribute. If the URL is dynamic, then JavaScript supports it. For example, you can present a dynamic personal page for a business contact on the Contact form. The following is an example of this JavaScript:

```
if (!is_new)
 form.linkedin_search.navigate = "http://www.linkedin.com/pub/dir/?last=" +
form.item['Last Name'] + "&first=" + form.item['First Name'];
```

If the url attribute is not set, then Oracle CRM On Demand Desktop loads the about:blank page by default.

XML Code to Customize Toolbars

This topic describes the code of the toolbars.xml file. It includes the following information:

- *Example Code of the toolbars.xml File*
- *Toolbars Element of the toolbars.xml File*

For more information, see *Customizing a Toolbar*.

Example Code of the toolbars.xml File

This topic gives one example of code that you can use in the toolbars.xml file. You might use this feature differently, depending on your business model. The following code is an example of the toolbars.xml file:

```
<button id="meeting_with_contact" name="#btn_meeting_with_contact"
small_image="orcl_meeting_with_contact:16">
 <action class="scriptable" id="meeting_with_contact"/>
</button>
```

Toolbars Element of the toolbars.xml File

The toolbars element is the root element of the toolbars.xml file. It describes the toolbar that Oracle CRM On Demand Desktop adds to a native Microsoft Outlook form or Microsoft Outlook window. The caption attribute of the toolbars element defines the caption of the toolbar. The toolbars element includes the following tags:

- **for.** Determines if Oracle CRM On Demand Desktop displays this element in a Microsoft Outlook window or Microsoft Outlook form. This depends if the element displayed is the explorer element or inspector element.
- **button.** A description of a button on a toolbar. For more information, see *Button Element of the Toolbars Element of the toolbars.xml File*.

Button Element of the Toolbars Element of the toolbars.xml File

The button element includes the following attributes:

- **name.** The caption for the button.
- **small_image.** The resource ID of the icon that Oracle CRM On Demand Desktop uses as the small icon for this button.

- **large_image.** The resource ID of the icon that Oracle CRM On Demand Desktop uses as a large icon for this button.
- **begin_group.** Determines if Oracle CRM On Demand Desktop displays the separator of the toolbar button for this button. It is useful if you must group toolbar buttons.

Action Element of the Button Element

The button element includes the action element, which defines the action that Oracle CRM On Demand Desktop calls if the user clicks the button. You can use a predefined action or write a custom action. You must set this action in the class attribute of the action element.

The action element includes the attachment element. To define attachment settings, you must define the following attributes of the attachment element:

- **type.** The type of the attachment object.
- **name_field.** The name of the field of the attachment object where Oracle CRM On Demand Desktop stores the name of the file.
- **body_field.** The name of the field of the attachment object where Oracle CRM On Demand Desktop stores the body of the file.
- **linking_field.** The name of the field of the attachment object where Oracle CRM On Demand Desktop stores the reference to the parent object.

Example Code of the Action Element of the Scriptable Action

Oracle CRM On Demand Desktop supports the scriptable action class. The action element of the scriptable action includes the id attribute. You can use the id attribute in a script to specify the action to perform. The following example specifies a button with a scriptable action:

```
<button id="new_account" name="#btn_new_account">
 <action class="scriptable" id="new_account"/>
</button>
```

In this example, Oracle CRM On Demand Desktop passes the value for the new_account attribute to the script when it handles the click event of the button. The script includes the predefined logic which the new_account attribute starts.

XML Code to Customize Dialog Boxes

This topic describes the code of the dialogs.xml file. It includes the following information:

- *Dialog Element of the dialogs.xml File*
- *Layout Element of the dialogs.xml File*
- *Appearance Element of the dialogs.xml File*

For more information, see *Customizing a Dialog Box*.

Dialog Element of the dialogs.xml File

The dialog element describes each dialog box. This element is similar to the form element of the forms_12.xml file, but it does not support the on_saved and on_saving attributes. The dialog element in the dialogs.xml file behaves in the same way as the form element in the forms_xx.xml file, except for the following differences:

- The dialog box description includes the layout element and the appearance element.
- The dialogs.xml file does not contain a validation_rules element.
- You cannot use native Microsoft Outlook controls in the dialogs.xml file.

Layout Element of the dialogs.xml File

The layout element is similar to the page element of the forms_12.xml file, but it includes different attributes. These attributes include:

- **sizable.** Determines if the user can change the size of the dialog box.
- **visible.** Sets the visibility for the dialog box during creation. If the visible attribute is set to False, then Oracle CRM On Demand Desktop creates the dialog box in the background. To make it visible, you must use JavaScript code to change the value for this attribute.
- **caption.** Defines the caption for the dialog box.
- **small_icon.** Defines the icon that displays next to the caption. The value of this attribute must be the ID of an image resource from Oracle CRM On Demand Desktop.

Appearance Element of the dialogs.xml File

The appearance element defines the position and size of the dialog box. It includes the following attributes:

- **height.** Defines the height of the dialog box, in pixels.
- **width.** Defines the width of the dialog box, in pixels.
- **position.** Defines the position of the dialog box in a screen. The position attribute can contain the following values:
  - **parent_center.** Displayed on the center of a parent window.
  - **desktop_center.** Displayed on the center of the desktop.
  - **custom.** A position that the user chooses.
  - **top.** The top position of the dialog box. The position attribute must include a custom value.
  - **left.** The left position of the dialog box. The position attribute must include a custom value.

XML Code to Customize Views Used for Microsoft Outlook Explorer List

This topic describes the code of the views.xml file. This file contains the set of str tags that define the configuration of the Microsoft Outlook view. The attribute of this element defines the unique name, or ID, for the view. For more information, see *Customizing a View*.

The following code is an example of the views.xml file:

```
<res_root>
 <str key="sample_view">
 <![CDATA[<?xml version="1.0"?>
 <view>
 <viewname>Sample view</viewname>
 <linecolor>8421504</linecolor>
 <linestyle>3</linestyle>
 <gridlines>1</gridlines>
 <newitemrow>0</newitemrow>
 <usequickflags>0</usequickflags>
 <collapsestate/>
 <previewstyle>color:Blue</previewstyle>
 <arrangement>
 <autogroup>0</autogroup>
 <collapseclient/>
 </arrangement>
 <column>
 <name>HREF</name>
 <prop>DAV:href</prop>
 <checkbox>1</checkbox>
 </column>
 <column>
 <maxrows>4294901760</maxrows>
 <heading>Organization</heading>
 <prop>urn:schemas:contacts:sn</prop>
 <type>string</type>
 <width>987</width>
 <style>padding-left:3px;;text-align:left</style>
 <editable>1</editable>
 <userheading>Organization</userheading>
 </column>
 <orderby>
 <order>
 <heading>Organization</heading>
 <prop>urn:schemas:contacts:sn</prop>
 <type>string</type>
 <userheading>Organization</userheading>
 <sort>asc</sort>
 </order>
 </orderby>
 <multiline>
 <width>0</width>
 </multiline>
 </view>]]>
 </str>
</res_root>
```

XML Code to Customize Views Used for Objects Forms

The view structure described in *XML Code to Customize Views Used for Microsoft Outlook Explorer List* is used to represent top-level objects displayed on the left side list of Outlook. Objects shown on UI forms in view controls must use the following structure:

```
<view id ="Custom view name to be used forms_XX.xml file">
 <image_list>
 <res_id type="normal">Icon name</res_id>
 </image_list>
 <columns>
 <column width="Specify column width" sort = "Specify sorting order: asc or desc">
 <heading type="string">Reference heading name from the package_res.xml file</heading>
 <field>Object field name</field>
 </column>
 </columns>
</view>
```

By using the above XML structure, you must list all of the columns that you want to display on the view, using the <column>…</column> tags. It is recommended that you do not change the above values for some of the attributes, for example, normal for the attribute type, and set the sorting order for no more than one field.

Note: If you are customizing the view either for association, or if you want some of the fields from the parent object to be displayed on the view, then you must use the extended format of the <field> element, as follows.

```
<field>
 <poly_join link="Id field that links objects">
 <if_type name="Object name of the field below">
 <simple>Field name</simple>
 </if_type>
 </poly_join>
</field>
```

Views for associations all contain extended structure for the <field> element. For example, this is the view representation for the Contact object, which is a part of many associations, such as Account.Contact.Association, Activity.Contact.Association, and so on. The views for associations are displayed on many objects UI forms, as follows:

```
<view id = "contacts:form_view">
 <image_list>
 <res_id type = "normal">type_image:Contact:16</res_id>
 </image_list>
 <columns>
 <column width = "126">
 <heading type = "string">head_first_name</heading>
 <field>
 <poly_join link="ContactId">
 <if_type name="Contact">
 <simple>ContactFirstName</simple>
 </if_type>
 </poly_join>
 </field>
 </column>
 <column sort = "asc" width = "135">
 <heading type = "string">head_last_name</heading>
 <field>
 <poly_join link="ContactId">
 <if_type name="Contact">
 <simple>ContactLastName</simple>
 </if_type>
 </poly_join>
 </field>
 </column>
 <column width = "132">
 <heading type = "string">head_email</heading>
 <field>
 <poly_join link="ContactId">
 <if_type name="Contact">
 <simple>ContactEmail</simple>
 </if_type>
 </poly_join>
 </field>
 </column>
 ...
 </columns>
</view>
```

It is recommended that you indicate where the view is to be used by using the following template:

```
<view id="<name of the view>:form_view|form_view-mvg|mvg|salesbook"></view>
```

For example, in the following line indicates a form-view is to be used for the opportunity_competitors view:

```
<view id="opportunity_competitors:form_view"></view>
```

Note: When you register a custom view, you must specify the `"custom_view_ctrl": true` option, as shown in the following example.

```
register_view_control_with_button(ctx, "Opportunity.Competitor",
"opportunity_competitor_view", "btn_add_opportunity_competitor",
{"custom_view_ctrl": true});
```

XML Code to Customize the Sales Book Control

This topic describes the code of the lookup_view_defs.xml file. It includes the following information:

- *Example Code of the lookup_view_defs.xml File*
- *Array Element of the lookup_view_defs.xml File*
- *Lookup View Definition Element of the lookup_view_defs.xml File*

For more information on configuring options for the SalesBook control in Microsoft Outlook, see *Customizing the SalesBook Control*.

Example Code of the lookup_view_defs.xml File

This topic gives one example of code that you can use in the lookup_view_defs.xml file. You might use this feature differently, depending on your business model. The following code is an example of the lookup_view_defs.xml file:

```
<res_root>
 <array key="all_lookup_types">
 <item value="Account"></item>
 <item value="Contact"></item>
 <item value="Opportunity"></item>
 </array>
 <lookup_view_def key="lookup:contacts">
 <display name="Contacts"></display>
 <filter dasl="[http://schemas.microsoft.com/mapi/proptag/0x001A001E] &gt;=
'IPM.Contact.OnDemand.Contact' AND [http://schemas.microsoft.com/mapi/proptag/
0x001A001E] &lt;= 'IPM.Contact.OnDemand.Contact'"></filter>
 <view id="contacts:salesbook"></view>
 <quick_lookup dasl_format="[http://schemas.microsoft.com/mapi/id/{00062004-
0000-0000-C000-000000000046}/8005001E] = '%s'"></quick_lookup>
 <type id="Contact"></type>
 </lookup_view_def>
</res_root>
```

To use the recommended custom SalesBook control, which is the default, you do not require the lookup view definitions listed in lookup_view_defs.xml. Instead, add selectors_options to business_logic.js in the following format:

```
scheme.objects.get_object("Account").selectors_options = {
 "source": {
 "caption": "",
 "allow_new": false,
 "view_id": "",
 "search_by": [""],
 "data_source_type": "",
 "online": {
 "view_id": "contacts:online_salesbook",
 "like_template": "*{keyword}*"
 }
 }
}
```

where:

- **get_object.** Indicates the name of the object for which you are adding selector options.
- **caption.** The caption of the object that is displayed on the SalesBook dialog box. You must specify the caption as a reference name from package_res.xml.
- **allow_new.** Enables or disables the Add button on the SalesBook dialog box. Possible values are True or False. If this parameter is not specified, then it is set to true by default.
- **view_id.** The identifier of the view defined in the views.xml file, which is displayed on the SalesBook dialog box.
- **search_by.** The name of the fields to search for in the SalesBook dialog box. You can enter several field names by separating them with a comma as in this example:

  ```
  "search_by": ["First Name", "Last Name"]
  ```

- **data_source_type.** Specifies where the object is stored. If the object is defined in the database section of the mapping XML file, then set data_source_type to `database`. If it is located in the types section, then set data_source_type to `outlook`.
- **"online": {"view_id".** The identifier of the view defined in the views.xml file that is displayed on the online SalesBook dialog box. If you use the same view for the Salesbook dialog box and the online SalesBook dialog box, then you must not specify this parameter.

If you create a direct or multi-value group (MVG) link using either of the `add_mvg_link` or `add_direct_link` functions, then pass null values instead of the view_id or left_objs_view_id or right_objs_view_id parameters.

Array Element of the lookup_view_defs.xml File

The array element defines a set of types that is available for the SalesBook control. The user cannot use the SalesBook control to choose a certain object type until you describe this type in the array element. Also, you must specify the type ID as a value attribute of the item element.

The following code is an example of the array element:

```
<array key="all_lookup_types">
 <item value="Account"></item>
 <item value="Contact"></item>
 <item value="Opportunity"></item>
</array>
```

Lookup View Definition Element of the lookup_view_defs.xml File

The lookup_view_def element describes the configuration for the SalesBook control. You can define as many configurations as you require. The key attribute defines the unique ID, or name, for this configuration.

The lookup_view_def element includes the following tags:

- **display.** The name attribute of the display element defines the name of this configuration. Oracle CRM On Demand Desktop displays it as a list value in the top-right corner of the SalesBook control.
- **filter.** The dasl attribute of the filter element describes the dasl filter that Oracle CRM On Demand Desktop applies to all objects that the array element describes. The user can view only the objects that match this filter in the SalesBook control.
- **view.** The id attribute of the view element defines the view that Oracle CRM On Demand Desktop applies to the list in the SalesBook control. You must describe this view ID in the views.xml file.
- **quick_lookup**. The dasl_format attribute of the quick_lookup element defines the dasl filter that Oracle CRM On Demand Desktop applies to the quick search feature of the SalesBook control. To simplify finding a field, this feature allows the user to enter any text to filter records. The user enters text in the top-left edit box on a SalesBook form.

  The following example code allows the user to view records where the File As field is the same as the string:

  ```
  <quick_lookup dasl_format="[http://schemas.microsoft.com/mapi/id/{00062004-
  0000-0000-C000-000000000046}/8005001E] = '%s'"></quick_lookup>
  ```

  where:

  - The File As field is ([http://schemas.microsoft.com/mapi/id/{00062004-0000-0000-C000-000000000046}/8005001E]).
  - The quick search is entered as ('%s').
- **type.** The ID attribute of the type element defines the type of object that Oracle CRM On Demand Desktop creates if the user clicks New in a SalesBook control. If you do not define this attribute, then the user cannot create a new object.

XML Code for Meta Information

This topic describes the code of the od_meta_info.xml file. It includes the following information:

- *SiebelMetaInfo Element of the od_meta_info.xml File*
- *Common_settings Element of the od_meta_info.xml File*
- *Object Element of the od_meta_info.xml File*
- *Field Element of the od_meta_info.xml File*
- *Extra_command_options Element of the od_meta_info.xml File*
- *Open_with_url_tmpl Element of the od_meta_info.xml File*
- *Picklist Element of the od_meta_info.xml File*
- *Master_filter_expr Element of the od_meta_info.xml File*

For more information, see *Customizing Meta Information*.

SiebelMetaInfo Element of the od_meta_info.xml File

The SiebelMetaInfo element is a root element. It does not contain attributes.

Common_settings Element of the od_meta_info.xml File

The common_settings element does not contain tags. However, it can contain subtags that you can use to specify common options for the Web Service connector. Oracle CRM On Demand Desktop supports the following subtags:

- **max_commands_per_batch.** Defines the maximum number of commands that Oracle CRM On Demand Desktop can place in a single batch. If Oracle CRM On Demand Desktop cannot interpret the value of this element as a positive integer value, then it does not apply any restrictions on the number of commands.
- **max_ids_per_command.** Defines the maximum number of object IDs that Oracle CRM On Demand Desktop can specify in a search specification for each independent request when a user performs a query by ID. It is the maximum number of record IDs that can be related to the parent record. For example:

```
<common_settings>
 <max_commands_per_batch>50</max_commands_per_batch>
 <max_ids_per_command>50</max_ids_per_command>
</common_settings>
```

Object Element of the od_meta_info.xml File

You can use the object element to define an object type that Oracle CRM On Demand Desktop supports. The following table describes the tags that Oracle CRM On Demand Desktop supports.

Element	Type	Description
EnableGetIDsBatching	Binary: yes or no	Allows or disallows you to obtain a batch of record Ids by using a single get command. The following values are valid: • **False.** Disallows you to obtain a batch of record Ids by using a single get command. • **True.** Allows you to obtain a batch of record Ids by using a single get command.
IntObjName	String	The name of the integration object that Oracle CRM On Demand Desktop uses for requests.
IsAssociation	Binary: yes or no	Indicates if this type of object is an association object.
IsCascadeDelete	Binary: yes or no	Not currently used.
IsTopLevel	Binary: yes or no	Indicates if a request for this type of object must be wrapped in a request for an object of some parent type.
Label	String	The label that Oracle CRM On Demand Desktop uses for this type of object in the user interface.
LabelPlural	String	The plural label that Oracle CRM On Demand Desktop uses for this type of object in the user interface.
SiebMsgXmlCollectionElemName	String	The name of the collection XML element that Oracle CRM On Demand Desktop uses in an Oracle CRM On Demand message.
SiebMsgXmlElemName	String	The name of the XML element that Oracle CRM On Demand Desktop uses in an Oracle CRM On Demand message.
SyncFrequency	Numeric	Identifies how often, Oracle CRM On Demand Desktop synchronizes the type, measured in seconds. For example, a setting of 30, sets the synchronization to take place every 30 seconds. If you set SyncFrequency to 0, then Oracle CRM On Demand Desktop synchronizes the type during every synchronization session. If you enter a positive integer, then Oracle CRM On Demand Desktop queries the Oracle CRM On Demand server for the records of this type in the time interval you define, starting from the time when the type was last queried.
TypeId	String	Unique ID of this type of object.
UpsertBusObjCacheSize	Numeric	Request attribute that defines the preferred cache size for each upsert operation for each object type. Oracle CRM On Demand uses this information. The following values are valid: • **5.** Default value for all types.

Element	Type	Description
		- **0.** Special value that you can use to resolve a problem that might exist with primaries.
ViewMode	String	Default ViewMode for this type of object.

Field Element of the od_meta_info.xml File

You can use the field element to define an object field. You must nest the field element in the definition of an object type. The following table describes the tags that Oracle CRM On Demand Desktop supports.

Element	Type	Description
DataType	String	Indicates the data type. The Web Service connector uses the value for this attribute to perform data conversion.
HasPicklist	Binary: yes or no	Indicates if this field is a bounded list.
IOElemName	String	Name of the XML element that Oracle CRM On Demand Desktop uses in Oracle CRM On Demand messages for values from this field.
IsCompositeId	Binary: yes or no	Not currently used.
IsFake	Binary: yes or no	If the value for this element is Yes, then Oracle CRM On Demand Desktop does not use the value from this field in any requests to the API.
IsFilterable	Binary: yes or no	Indicates if this field is available to choose a filter expression on the control panel.
IsMVGField	Binary: yes or no	Not currently used.
IsNullable	Binary: yes or no	The synchronization engine uses this element to break a circular reference.
IsPartOfUserKey	Binary: yes or no	Indicates if this field is part of a user key.
IsPrimaryKey	Binary: yes or no	Indicates if Oracle CRM On Demand Desktop uses the value from this field as the primary key for the object. Only one field on an object type can be marked as the primary key.
IsReadonly	Binary: yes or no	Not currently used.

Element	Type	Description
IsRefObjId	Binary: yes or no	Indicates if Oracle CRM On Demand Desktop uses this element as a foreign key field. The Web Service connector and the synchronization engine use this element.
IsRequired	Binary: yes or no	Not currently used.
IsTimestamp	Binary: yes or no	Indicates if Oracle CRM On Demand Desktop uses the value from this field as an object timestamp. You can define only one timestamp for each type of object.
Label	String	The label that is displayed in the user interface.
Name	String	Unique name of the field. If the IsFake element for this field is set to No, and if this field is not present in requests to the Oracle CRM On Demand server, then this name must be identical to the field name of the Integration Component from the API
OrderNumber	Numeric	Indicates the order number of this field. The Web Service connector uses this element internally. If several fields are defined that hold a reference to the Oracle CRM On Demand parent record, then their order numbers must reflect the nesting order of the parent types. This situation occurs if an association exists between two parent types where one of the parent types is nested in the other parent type.
PicklistCollectionType	String	Type of list items.
PicklistIsStatic	Binary: yes or no	Indicates if the associated list is static or dynamic.
PicklistTypeId	String	The name of the type of the list object. You must use the picklist element to define the list in the od_meta_info.xml file.
RefObjIsParent	Binary: yes or no	Indicates if the object type that is referenced is a parent. The Web Service connector uses this element connector to build the hierarchy for the object type.
RefObjTypeId	String	The name of the object type to which this field refers. This object type must be defined in the od_meta_info.xml file.

Extra_command_options Element of the od_meta_info.xml File

You can use the extra_command_options element in the definition of an object type to specify extra options that Oracle CRM On Demand Desktop passes to a command element on each request. You can use the option subtag with the following tags to specify each element:

- **Name.** The name of the extra command attribute

- **Value.** The value of the extra command attribute

Open_with_url_tmpl Element of the od_meta_info.xml File

You can use the open_with_url_tmpl element in the definition of an object type. You use this element to specify a template that Oracle CRM On Demand Desktop uses to open records of this object type in Oracle CRM On Demand.

The open_with_url_tmpl element includes the following format:

```
<open_with_url_tmpl>
 <![CDATA[
 "URL template"
 ]]>
</open_with_url_tmpl>
```

Oracle CRM On Demand Desktop uses macros in the CDATA section for the attributes that Oracle CRM On Demand Desktop uses in the open_with_url_tmpl template. Each attribute includes the following format:

```
:[:(attribute):]
```

A series of attributes in the command uses the following format:

```
:[:(protocol):]://:[:(hostname):]:[:(port):]:[:(own_id):]
```

where:

- protocol is automatically replaced with the URL protocol. The value is http or https.
- hostname is automatically replaced with the name of the Oracle CRM On Demand server.
- port is automatically replaced with the port number of the Oracle CRM On Demand server.
- own_id is automatically replaced with the object ID that Oracle CRM On Demand Desktop uses to open Oracle CRM On Demand.

For each macro in the command, Oracle CRM On Demand Desktop replaces the macro with the value at run time. The definition of the template must be enclosed in a CDATA construct. For example, you can use the following code for account objects:

```
<open_with_url_tmpl>
 <![CDATA[
 :[:(protocol):]://:[:(hostname):]::[:(port):]/OnDemand/user/
AccountDetail?AccountDetailForm.Id=:[:(own_id):]
 ]]>
</open_with_url_tmpl>
```

Picklist Element of the od_meta_info.xml File

You can use the picklist element to define a static list. The following table describes the tags that Oracle CRM On Demand Desktop supports.

Element	Type	Description
TypeID	String	Unique name for the list.

Element	Type	Description
SrcObjectTypeId	String	Name of the object type that Oracle CRM On Demand Desktop uses to retrieve items for this list. This type must be defined in the od_meta_info.xml file.
CollectionTypeFldName	String	Name of the field on the original object that contains the type for the list items.
ValueFldName	String	Name of the field on the original object from which Oracle CRM On Demand Desktop retrieves values for the list items.
LabelFldName	String	Name of the field on the original object from which Oracle CRM On Demand Desktop retrieves labels for the list items.
LangFldName	String	Name of the field on the original object that contains the language code.

Master_filter_expr Element of the od_meta_info.xml File

You can nest the master_filter_expr element in the definition of a list. You can use this element to define a filter expression that Oracle CRM On Demand Desktop applies to any request for items of this list. You must enclose the value for this element with a CDATA section and display the values for the search specification attributes of the target object type.

13 Additional Code in the Customization Example

Additional Code in the Customization Example

This chapter includes the more lengthy XML code that the customization example requires. It includes the following topics:

- *XML Code That Defines a Set of Custom Fields*
- *XML Code That Maps a Definition for a Custom Object*
- *XML Code That Hides Unused Microsoft Outlook Controls*
- *XML Code of a Complete Form for Custom Object1*
- *XML Code That Defines a Many-To-Many Relationship*
- *XML Code That Defines a List*
- *XML Code That Creates Cells*

XML Code That Defines a Set of Custom Fields

To define a set of custom fields, you add the following code to the type element of the od_basic_mapping.xml file:

```
<type id="ChannelPartner" hidden_folder="true" folder_type="10"
display_name="CHPT">
 <form message_class="IPM.Contact.OnDemand.Channel_Partner"
display_name="Channel Partner" icon="type_image:User:16"></form>
 <field id="Name">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A110000"></mapi_tag>
 <convertor><string/></convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="LastName"></outlook_field>
 <convertor><string/></convertor>
 </outlook_std>
 </writer>
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A060000"></mapi_tag>
 <convertor><string/></convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="FirstName"></outlook_field>
 <convertor><string/></convertor>
 </outlook_std>
 </writer>
 </field>
 <field id="Location">
 <reader>
 <mapi_user>
 <user_field id="od Location" ol_field_type="1"></user_field>
 <convertor><string/></convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="od Location" ol_field_type="1"></user_field>
 <convertor><string/></convertor>
 </outlook_user>
 </writer>
 </field>
</type>
```

XML Code That Maps a Definition for a Custom Object

The following code is an example of a complete mapping definition for Custom Object 1 in the od_basic_mapping.xml file:

```
<types>
 <type id="Custom Object 1" folder_type="10" display_name="CustomObject1" ver="2">
 <form message_class="IPM.Contact.OnDemand.CustomObject1"
display_name="CustomObject1" icon="type_image:Generic:16">OnDemand CustomObject1</form>
 <custom_views default_name = "#view_od_co1">
 <view id="all_customobject1" name ="#view_od_co1"> </view>
 </custom_views>
 <field id="Name">
 <reader>
 <mapi_user>
 <user_field id="od Name" ol_field_type="1"></user_field>
 <convertor>
 <string/>
 </convertor>
 </mapi_user>
 </reader>
 <writer>
 <outlook_user>
 <user_field id="od Name" ol_field_type="1"></user_field>
 <convertor>
 <string/>
 </convertor>
 </outlook_user>
 </writer>
 </field>
 <field id="Description">
 <reader>
 <mapi_std>
 <mapi_tag id="0x3A110000"></mapi_tag>
 <convertor>
 <string/>
 </convertor>
 </mapi_std>
 </reader>
 <writer>
 <outlook_std>
 <outlook_field id="LastName"></outlook_field>
 <convertor>
 <string/>
 </convertor>
 </outlook_std>
 </writer>
 </field>
 </type>
</types>
```

If the object you are adding is defined in CRM Desktop storage, then you must add the following object definition to the <database> element, which you can find in the od_basic_mapping.xml file:

```
<type id="Custom Object 1" icon="type_image:Custom Object 1:16">
 <field id ="Name">
 <type>
 <simple type="string"/>
 </type>
 </field>
 <field id ="Description">
 <type>
 <simple type="string"/>
 </type>
 </field>
</type>
```

XML Code That Hides Unused Microsoft Outlook Controls

The following code is an example of how the unused native Microsoft Outlook controls are removed in to a small cell:

```
<form id="OnDemand CustomObject1" >
 <page id="General" tag="0x10A6" min_height="155" min_width="520">
 ...
 ...
 <!-- hidden section -->
 <cell size="0">
 <stack layout="vert">
 <cell>
 <control id="0x103f" window_id="0x103f"></control>
 <!-- Description -->
 </cell>
 <cell>
 <control id="0x6f2a" window_id="0x6f2a"></control>
 <!-- Full Name button -->
 </cell>
 <cell>
 <control id="0x1000" window_id="0x1000"></control>
 <!-- Full Name edit -->
 </cell>
 <cell>
 <control id="0x11a2" window_id="0x11a2"></control>
 <!--Company static-->
 </cell>
 <cell>
 <control id="0x1181" window_id="0x1181"></control>
 <!-- Company edit -->
 </cell>
 <cell>
 <control id="0x11a3" window_id="0x11a3"></control>
 <!-- Job Title static -->
 </cell>
 <cell>
 <control id="0x1180" window_id="0x1180"></control>
 <!-- Job Title edit -->
 </cell>
 <cell>
```

```
<control id="0x11a4" window_id="0x11a4"></control>
<!-- FileAs static -->
</cell>
<cell>
<control id="0x1182" window_id="0x1182"></control>
<!-- FileAs combo -->
</cell>
<cell>
<control id="0x11a1" window_id="0x11a1"></control>
<!--phone numbers separator-->
</cell>
<cell>
<control id="0x11a5" window_id="0x11a5"></control>
<!--Phone numbers static-->
</cell>
<cell>
<control id="0x1113" window_id="0x1113"></control>
<!-- Business phone button -->
</cell>
<cell>
<control id="0x110a" window_id="0x110a"></control>
<!-- Business Phone type (small) button -->
</cell>
<cell>
<control id="0x1001" window_id="0x1001"></control>
<!-- Business Phone Phone edit -->
</cell>
<cell>
<control id="0x110b" window_id="0x110b"></control>
<!-- Home Phone Type button -->
</cell>
<cell>
<control id="0x1002" window_id="0x1002"></control>
<!-- Home Phone edit -->
</cell>
<cell>
<control id="0x1114" window_id="0x1114"></control>
<!-- Home Phone button -->
</cell>
<cell>
<control id="0x110c" window_id="0x110c"></control>
<!-- Business Fax Type button -->
</cell>
<cell>
<control id="0x1003" window_id="0x1003"></control>
<!-- Business Fax edit -->
</cell>
<cell>
<control id="0x1115" window_id="0x1115"></control>
<!-- Business Fax button -->
</cell>
<cell>
<control id="0x1116" window_id="0x1116"></control>
<!--Mobile phone button -->
</cell>
<cell>
<control id="0x110d" window_id="0x110d"></control>
<!-- Mobile Phone Type button -->
</cell>
<cell>
<control id="0x1004" window_id="0x1004"></control>
<!-- Mobile Phone edit -->
</cell>
<cell>
<control id="0x11a7" window_id="0x11a7"></control>
<!-- Addresses section header -->
```

```
</cell>
<cell>
<control id="0x11a8" window_id="0x11a8"></control>
<!--Address separator -->
</cell>
<cell>
<control id="0x6f2b" window_id="0x6f2b"></control>
<!-- Address (Business...) button -->
</cell>
<cell>
<control id="0x1109" window_id="0x1109"></control>
<!-- Address Type button -->
</cell>
<cell>
<control id="0x1080" window_id="0x1080"></control>
<!-- Mailing Address checkbox -->
</cell>
<cell>
<control id="0x1017" window_id="0x1017"></control>
<!--Address edit -->
</cell>
<cell>
<control id="0x1108" window_id="0x1108"></control>
<!-- Contact Picture button -->
</cell>
<cell>
<control id="0x11c7" window_id="0x11c7"></control>
<!--Business Card -->
</cell>
<cell>
<control id="0x11bb" window_id="0x11bb"></control>
<!-- Internet caption-->
</cell>
<cell>
<control id="0x11c5" window_id="0x11c5"></control>
<!-- Internet separator-->
</cell>
<cell>
<control id="0x1101" window_id="0x1101"></control>
<!-- Email AddressBook button -->
</cell>
<cell>
<control id="0x111b" window_id="0x111b"></control>
<!--email address button small -->
</cell>
<cell>
<control id="0x1018" window_id="0x1018"></control>
<!--email address edit -->
</cell>
<cell>
<control id="0x11c6" window_id="0x11c6"></control>
<!-- DisplayAs static -->
</cell>
<cell>
<control id="0x101c" window_id="0x101c"></control>
<!-- DisplayAs edit -->
</cell>
<cell>
<control id="0x11aa" window_id="0x11aa"></control>
<!-- Web Page static -->
</cell>
<cell>
<control id="0x11a9" window_id="0x11a9"></control>
<!-- Web Page edit -->
</cell>
<cell>
```

```
 <control id="0x11ba" window_id="0x11ba"></control>
 <!--IM address static-->
 </cell>
 <cell>
 <control id="0x1016" window_id="0x1016"></control>
 <!-- IM Address edit -->
 </cell>
 <cell>
 <control id="0x11a6" window_id="0x11a6"></control>
 <!-- Notes Caption -->
 </cell>
 <cell>
 <control id="0x11b7" window_id="0x11b7"></control>
 <!-- Notes separator -->
 </cell>
 <cell>
 <control id="0x200" window_id="0x200"></control>
 <!-- "In folder" caption -->
 </cell>
 <cell>
 <control id="0x201" window_id="0x201"></control>
 <!-- Folder icon -->
 </cell>
 <cell>
 <control id="0x202" window_id="0x202"></control>
 <!-- Folder name -->
 </cell>
 <cell>
 <control id="0x10A3" window_id="0x10A3"></control>
 <!-- Contacts botton -->
 </cell>
 <cell>
 <control id="0x10A4" window_id="0x10A4"></control>
 <!-- Contacts Edit -->
 </cell>
 <cell>
 <control id="hidden_address_map_it" window_id="0x111c"></control>
 <!-- Map it button -->
 </cell>
 <cell>
 <control id="hidden_address_static" window_id="0x11C3"></control>
 <!-- Static control -->
 </cell>
 <!-- Contact picture -->
 <cell>
 <control id="0x1023" window_id="0x1023"/>
 </cell>
 </stack>
 </cell>
 </page>
</form
```

XML Code of a Complete Form for Custom Object1

The following code is an example of the complete custom form for Custom Object 1 in the forms_12.xml file:

```
<form id="OnDemand CustomObject1" >
 <script>
 <![CDATA[
 include("forms.js", "forms");
 var ctx = {
```

```
"application": application,
"ui": ui,
"application_script": application_script,
"form": form
};
var current_form = new forms.od_co1_form(forms.create_form_ctx(ctx));
]]>
</script>
<page id="General" tag="0x10A6" min_height="155" min_width="520">
<cell>
<stack layout="vert" padding="5">
<cell>
<stack layout="horz" spacing="3">
<cell size="65">
<stack layout="vert" spacing="5">
<cell size="21">
<static id="lbl_co1_name" tab_order="2">
<text>#lbl_co1_name</text>
</static>
</cell>
<cell size="21">
<static id="lbl_co1_description" tab_order="4">
<text>#lbl_co1_description</text>
</static>
</cell>
</stack>
</cell>
<cell>
<stack layout="vert" spacing="5">
<cell size="21">
<edit id="Name" max_chars="100" tab_order="3">
<field value="string">Name</field>
</edit>
</cell>
<cell size="21">
<edit id="Description" max_chars="100" tab_order="5">
<field value="string">Description</field>
</edit>
</cell>
</stack>
</cell>
<!-- hidden section -->
<cell size="0">
<stack layout="vert">
<cell>
<control id="0x103f" window_id="0x103f"></control>
<!-- Description -->
</cell>
<cell>
<control id="0x6f2a" window_id="0x6f2a"></control>
<!-- Full Name button -->
</cell>
<cell>
<control id="0x1000" window_id="0x1000"></control>
<!-- Full Name edit -->
</cell>
<cell>
<control id="0x11a2" window_id="0x11a2"></control>
<!--Company static-->
</cell>
<cell>
<control id="0x1181" window_id="0x1181"></control>
<!-- Company edit -->
</cell>
<cell>
<control id="0x11a3" window_id="0x11a3"></control>
```

```
<!-- Job Title static -->
</cell>
<cell>
<control id="0x1180" window_id="0x1180"></control>
<!-- Job Title edit -->
</cell>
<cell>
<control id="0x11a4" window_id="0x11a4"></control>
<!-- FileAs static -->
</cell>
<cell>
<control id="0x1182" window_id="0x1182"></control>
<!-- FileAs combo -->
</cell>
<cell>
<control id="0x11a1" window_id="0x11a1"></control>
<!--phone numbers separator-->
</cell>
<cell>
<control id="0x11a5" window_id="0x11a5"></control>
<!--Phone numbers static-->
</cell>
<cell>
<control id="0x1113" window_id="0x1113"></control>
<!-- Business phone button -->
</cell>
<cell>
<control id="0x110a" window_id="0x110a"></control>
<!-- Business Phone type (small) button -->
</cell>
<cell>
<control id="0x1001" window_id="0x1001"></control>
<!-- Business Phone Phone edit -->
</cell>
<cell>
<control id="0x110b" window_id="0x110b"></control>
<!-- Home Phone Type button -->
</cell>
<cell>
<control id="0x1002" window_id="0x1002"></control>
<!-- Home Phone edit -->
</cell>
<cell>
<control id="0x1114" window_id="0x1114"></control>
<!-- Home Phone button -->
</cell>
<cell>
<control id="0x110c" window_id="0x110c"></control>
<!-- Business Fax Type button -->
</cell>
<cell>
<control id="0x1003" window_id="0x1003"></control>
<!-- Business Fax edit -->
</cell>
<cell>
<control id="0x1115" window_id="0x1115"></control>
<!-- Business Fax button -->
</cell>
<cell>
<control id="0x1116" window_id="0x1116"></control>
<!--Mobile phone button -->
</cell>
<cell>
<control id="0x110d" window_id="0x110d"></control>
<!-- Mobile Phone Type button -->
</cell>
```

```
<cell>
<control id="0x1004" window_id="0x1004"></control>
<!-- Mobile Phone edit -->
</cell>
<cell>
<control id="0x11a7" window_id="0x11a7"></control>
<!-- Addresses section header -->
</cell>
<cell>
<control id="0x11a8" window_id="0x11a8"></control>
<!--Address separator -->
</cell>
<cell>
<control id="0x6f2b" window_id="0x6f2b"></control>
<!-- Address (Business...) button -->
</cell>
<cell>
<control id="0x1109" window_id="0x1109"></control>
<!-- Address Type button -->
</cell>
<cell>
<control id="0x1080" window_id="0x1080"></control>
<!-- Mailing Address checkbox -->
</cell>
<cell>
<control id="0x1017" window_id="0x1017"></control>
<!--Address edit -->
</cell>
<cell>
<control id="0x1108" window_id="0x1108"></control>
<!-- Contact Picture button -->
</cell>
<cell>
<control id="0x11c7" window_id="0x11c7"></control>
<!--Business Card -->
</cell>
<cell>
<control id="0x11bb" window_id="0x11bb"></control>
<!-- Internet caption-->
</cell>
<cell>
<control id="0x11c5" window_id="0x11c5"></control>
<!-- Internet separator-->
</cell>
<cell>
<control id="0x1101" window_id="0x1101"></control>
<!-- Email AddressBook button -->
</cell>
<cell>
<control id="0x111b" window_id="0x111b"></control>
<!--email address button small -->
</cell>
<cell>
<control id="0x1018" window_id="0x1018"></control>
<!--email address edit -->
</cell>
<cell>
<control id="0x11c6" window_id="0x11c6"></control>
<!-- DisplayAs static -->
</cell>
<cell>
<control id="0x101c" window_id="0x101c"></control>
<!-- DisplayAs edit -->
</cell>
<cell>
<control id="0x11aa" window_id="0x11aa"></control>
```

```
 <!-- Web Page static -->
 </cell>
 <cell>
 <control id="0x11a9" window_id="0x11a9"></control>
 <!-- Web Page edit -->
 </cell>
 <cell>
 <control id="0x11ba" window_id="0x11ba"></control>
 <!--IM address static-->
 </cell>
 <cell>
 <control id="0x1016" window_id="0x1016"></control>
 <!-- IM Address edit -->
 </cell>
 <cell>
 <control id="0x11a6" window_id="0x11a6"></control>
 <!-- Notes Caption -->
 </cell>
 <cell>
 <control id="0x11b7" window_id="0x11b7"></control>
 <!-- Notes separator -->
 </cell>
 <cell>
 <control id="0x200" window_id="0x200"></control>
 <!-- "In folder" caption -->
 </cell>
 <cell>
 <control id="0x201" window_id="0x201"></control>
 <!-- Folder icon -->
 </cell>
 <cell>
 <control id="0x202" window_id="0x202"></control>
 <!-- Folder name -->
 </cell>
 <cell>
 <control id="0x10A3" window_id="0x10A3"></control>
 <!-- Contacts botton -->
 </cell>
 <cell>
 <control id="0x10A4" window_id="0x10A4"></control>
 <!-- Contacts Edit -->
 </cell>
 <cell>
 <control id="hidden_address_map_it" window_id="0x111c"></control>
 <!-- Map it button -->
 </cell>
 <cell>
 <control id="hidden_address_static" window_id="0x11C3"></control>
 <!-- Static control -->
 </cell>
 <!-- Contact picture -->
 <cell>
 <control id="0x1023" window_id="0x1023"/>
 </cell>
 </stack>
 </cell>
 </stack>
 </cell>
 </stack>
 </cell>
 </page>
</form>
```

XML Code That Defines a Many-To-Many Relationship

Many-to-many relationships are typically defined using association objects. It is recommended that you place these objects in DB FACADE storage by adding the following XML code to the od_basic_mapping.xml file:

```
<type id="Opportunity.Channel_Partner.Association"
icon="type_image:Opportunity.Channel_Partner.Association:16">
 <field id="OpportunityId">
 <type>
 <simple type="binary" />
 </type>
 </field>
 <field id="ChannelPartnerId">
 <type>
 <foreign_key>
 <type_id>Channel_Partner</type_id>
 </foreign_key>
 </type>
 </field>
 <field id="ChannelPartnerStatus">
 <type>
 <simple type="string" />
 </type>
 </field>
</type>
```

XML Code That Defines a List

To define a drop-down list, you add the following code to the od_basic_mapping.xml file on the activity object:

```
<type id="ActionTypePicklist" icon="type_image:ActionTypePicklist:16">
 <field id="Label">
 <type>
 <simple type="string" />
 </type>
 </field>
 <field id="Value">
 <type>
 <simple type="string" />
 </type>
 </field>
 <field id="SortOrder">
 <type>
 <simple type="integer" />
 </type>
 </field>
 <field id="IsDefault">
 <type>
 <simple type="boolean" />
 </type>
 </field>
 <field id="Parent">
 <type>
 <simple type="string" />
 </type>
 </field>
```

```
</type>
```

XML Code That Creates Cells

To define part of the layout of the form, you add the following code to the forms_12.xml file:

```
<stack layout="horz" spacing="3">
 <cell size="5" attraction="far"></cell>
 <!-- left side captions -->
 <cell size="90">
 <stack spacing="5" layout="vert" padding="4">
 <cell size="21">
 <static id="lbl_account" tab_order="25">
 <text>#lbl_account</text>
 </static>
 </cell>
 <cell size="21">
 <static id="lbl_opportunity" tab_order="28">
 <text>#lbl_opportunity</text>
 </static>
 </cell>
 <cell size="21">
 <static id="lbl_lead" tab_order="31">
 <text>#lbl_lead</text>
 </static>
 </cell>
 </stack>
 </cell>
 <!-- left side fields -->
 <cell>
 <stack layout="vert" spacing="5">
 <cell size="21">
 <stack layout="horz" spacing="3">
 <cell>
 <autocomplete id="account_id" tab_order="26">
 <field>AccountId</field>
 <source type="Account" format=":[:(AccountName):]"></source>
 </autocomplete>
 </cell>
 <cell size="21" attraction="far">
 <button id="btn_account_select" image="lookup_button" tab_order="27">
 <text>...</text>
 </button>
 </cell>
 </stack>
 </cell>
 <cell size="21">
 <stack layout="horz" spacing="3">
 <cell>
 <scriptable_autocomplete id="opportunity_id" tab_order="29">
 <!--field>OpportunityId</field>
 <source type="Opportunity"
format=":[:(OpportunityName):]"></source-->
 </scriptable_autocomplete>
 </cell>
 <cell size="21" attraction="far">
 <button id="btn_opportunity_select" image="lookup_button" tab_order="30">
 <text>...</text>
 </button>
 </cell>
 </stack>
```

```
     </cell>
     <cell size="21">
     <stack layout="horz" spacing="3">
     <cell>
     <scriptable_autocomplete id="lead_id" tab_order="32">
     <!--field>LeadId</field>
     <source type="Lead" format=":[:(LeadFullName):]"></source-->
     </scriptable_autocomplete>
     </cell>
     <cell size="21" attraction="far">
     <button id="btn_lead_select" image="lookup_button" tab_order="33">
     <text>...</text>
     </button>
     </cell>
     </stack>
     </cell>
     </stack>
     </cell>
    </stack>
```

Related Topics

Process of Customizing Objects in Oracle CRM On Demand Desktop

14 Windows Registry Parameters You Can Use With Oracle CRM On Demand Desktop

Windows Registry Parameters You Can Use With Oracle CRM On Demand Desktop

This chapter describes Windows registry parameters that you can use with Oracle CRM On Demand Desktop. It includes the following topics:

- *Windows Registry Parameters That Affect Oracle CRM On Demand Desktop Behavior*
- *Windows Registry Parameters That Affect Local Synchronization*
- *Windows Registry Parameters That You Can Use with Log Files and Dump Files*

Windows Registry Parameters That Affect Oracle CRM On Demand Desktop Behavior

The following table describes the Windows Registry parameters that you can modify to change Oracle CRM On Demand Desktop behavior. In the Registry Editor (regedit), you can modify these parameters in the following path:

```
HKEY_CURRENT_USER\Software\Oracle\CRM Desktop
```

Windows Registry Parameter	Description
AppLanguageID	The ID of the current installation package of Microsoft Outlook.
ConnectorCfgPath	The path to the connector_configuration.xml file in the customization package.
DestinationProfile	The name of the Microsoft Outlook profile where Oracle CRM On Demand Desktop is installed.
DestinationStore	The location of the profile identified in the DestinationProfile.
DisableSyncConfirmation	Suppresses the confirmation for deleting an object. The following values are valid: • **1.** Suppress confirmation for deleting an object. • **0.** Confirmation for deleting an object, which is the default value. The corresponding attribute in the connector_configuration.xml file of the customization package automatically overwrites the DisableSyncConfirmation key.
FiltersEstimateOnTimer	Sets the interval in milliseconds for an automatic estimate of the records after the filters are changed in the control panel. The following values are valid:

Windows Registry Parameter	Description
	• **0.** Do not estimate automatically. • **1.** Estimate automatically. This entry is not accessible through the administrative interface.
HTTPClient:AcceptCompression	A flag that instructs the Web service connector to accept compressed HTTP content. This parameter is not accessible through the administrative interface.
HTTPClient:CompressOutgoing	A flag that instructs the Web service connector to send zipped HTTP content. This parameter is not accessible through the administrative interface.
HTTPClient:ConnectTimeout	The timeout period in milliseconds for the connection.
HTTPClient:ReceiveTimeout	The timeout period in milliseconds for receiving requests.
HTTPClient:SendRetryCount	The number of connection retries for a period of time in milliseconds.
HTTPClient:SendTimeout	The timeout period in milliseconds for sending requests.
Page:Feedback:AttachLog	Specifies the log file for the feedback form. The following values are valid: • **0.** Do not attach a log file to the feedback form. • **1.** Attach a log file to the feedback form.
ProxyLogin	The login name for the proxy server.
ProxyPassword	The password for the proxy server.
ProxyServer	The host name for the proxy server.
ProxyServerPort	The port number for the proxy server.
ProxyUsage	The flag that specifies a proxy server. The following values are valid: • **0.** Do not use a proxy server. • **1.** Use a proxy server.
RunPeriodicalSyncAlways	Determines if Oracle CRM On Demand Desktop starts a scheduled synchronization at the scheduled time or waits until Microsoft Outlook is idle. The following values are available: • **0.** Wait until Microsoft Outlook is idle to start the scheduled synchronization. • **1.** Start the scheduled synchronization immediately when the synchronization is scheduled to occur. This value is the default.
SessionsKeepAliveAmount	Defines the number of synchronization sessions to store in the internal database as history. Oracle CRM On Demand Desktop stores statistical information for each synchronization session. The user can use the list control in the Sync Issues tab of the Synchronization Control Panel to view information about synchronization issues.

Windows Registry Parameter	Description
SharedByDefault:NewItems	The SharedByDefault:NewItems registry key determines how Oracle CRM On Demand Desktop shares newly created Microsoft Outlook items. For more information, see *Overview of How Oracle CRM On Demand Desktop Handles an Activity*.
Connector:HideSavePasswordOption	Determines how Oracle CRM On Demand Desktop displays the Save Password check box on the login screen. The following values are valid: • **0.** Display the Save Password check box. By default, the check box is displayed. • **1.** Do not display the Save Password check box.
Connector:MetaInfoFilePath	Describes the path to the od_meta_info.xml file in the customization package.
Connector:Protocol	Defines the URL protocol. The default is http. The following values are valid: • http • https **Note:** Only https protocol is allowed in production environments.
Connector:RequestSuffix	Defines the suffix of the URL that points to the Oracle CRM On Demand server. The following is the default value: `Services/Integration`
Connector:Server	Defines the host name of the URL. The default is empty.
Connector:ServerPort	Defines the port of the URL. The default is 80 for http protocol and 443 for https protocol. Only https protocol is allowed in production environments.
SuppressSyncEstimating	At the beginning of the synchronization, this parameter determines how Oracle CRM On Demand Desktop gets an estimate of the number of all the object types that are visible to the current user. The following values are valid: • **0.** Counts of all the object types. • **1.** Do not count all the object types.
SwitchToAutoCompleteChoiceLimit	Limits the number of items that Oracle CRM On Demand Desktop displays in the box in the Filters tab when the user creates rules.
SyncDumpBaseFname	Defines the full path and file name for the synchronization dump file that Oracle CRM On Demand Desktop creates separately for each synchronization. Oracle CRM On Demand Desktop stores synchronization history information in this file. Oracle CRM On Demand Desktop uses it for investigation purposes.
Connector:SOAPDumpBaseFname	Describes the path to where Oracle CRM On Demand Desktop stores information about SOAP activity. This information describes the requests and replies of the Web Service Connector.

Windows Registry Parameters That Affect Local Synchronization

The following table describes the options in the Windows Registry that control local synchronization behavior. For more information, see *Modifying the Windows Registry Parameters That Affect Local Synchronization*.

Windows Registry Parameter	Description
EnablePeriodicUpstreamSync	Enables or disables local synchronization. The following values are valid: • **0.** Synchronization is disabled. • **1.** Synchronization is enabled.
UpstreamSyncIsDefault	Sets the default synchronization type. The following values are valid: • **0.** Full synchronization • **1.** Local synchronization
CustomSyncPeriodsUpstream	Specifies the intervals for local synchronization. The default values are: • `120 = page-sync-periods-2-hours` • `240 = page-sync-periods-4-hours` • `360 = page-sync-periods-6-hours` • `1440 = page-sync-periods-1-day` You must describe these values in the package_res.xml file as resource strings: • `<str key="page-sync-periods-2-hours">Once in two hours</str>` • `<str key="page-sync-periods-4-hours">Once in four hours</str>` • `<str key="page-sync-periods-6-hours">Once in six hours</str>` • `<str key="page-sync-periods-1-day">Once a day</str>`
SyncPeriodUpstream	Specifies the default value for the local synchronization interval in milliseconds. The default value is 00dbba00 (4 hours).
UpstreamSyncMinThreshold	Disables local synchronization if the general synchronization occurs more frequently than the specified value. The default value is 60000 milliseconds (6 hours).

Windows Registry Parameters That You Can Use with Log Files and Dump Files

This topic describes parameters that you can use with log files and dump files. It includes the following information:

- *Windows Registry Parameters That You Can Use with the General Log File*
- *Windows Registry Parameters That You Can Use with the Exception Log File*
- *Windows Registry Parameters That You Can Use with the Crash Dump File*
- *Windows Registry Parameters That You Can Use with the SOAP Dump File*
- *Windows Registry Parameters That You Can Use with the Synchronization Dump File*

An integer that you can use with a parameter represents a DWORD value in the registry in Windows 7 or a 32-bit DWORD value in the registry in Windows Vista.

Windows Registry Parameters That You Can Use with the General Log File

The following table describes the parameters that you can use with the general log.

Parameter	Description
enable	You can use one of the following values: • **1.** The general log is enabled. • **0.** The general log is disabled. To disable the general log file, you must use the registry. You cannot use the Logging Configuration dialog box. For more information, see *Assigning a Logging Profile to Oracle CRM On Demand Desktop*.
log_level	Defines the level of detail in the log files. You can use one of the following integers: • **4294967295** — Disables all. • **0** — Enables all. • **1000** — Shows debug information. • **2000** — Shows information messages. • **3000** — Shows warnings. • **4000** — Shows errors. • **5000** — Shows fatal errors.
out_file	You can include a string that identifies the file name of the general log file. For example, assume you use the following value: log.txt

Parameter	Description
	In this example, the first file name is log.0000.txt, the second file name is log.0001.txt, and so forth. If you do not include the out_file parameter, then Oracle CRM On Demand Desktop uses log.txt as the default value.
enable_dbg_window	You can use one of the following values: - **1.** Enables output through the OutputDebugString that Oracle CRM On Demand Desktop displays in the VS Debug Output window during a debugging session. The OutputDebugString is a string that Oracle CRM On Demand Desktop sends to the debugger for display. The VS Debug Output window is a window in Microsoft Visual Studio that displays debugging results. For more information about Visual Studio, see the documentation at the Microsoft TechNet Web site. - **0.** Disables the general log file.
enable_cout	You can use one of the following values: - **1.** Enables logging in the console. - **0.** Disables logging in the console.
max_size_bytes	You can use an integer that sets the maximum number of bytes that a single log file can contain. For example, if you set max_size_bytes to 10485760 bytes, and if the current log reaches 10485760 bytes, then Oracle CRM On Demand Desktop creates a new log file. This behavior is similar to setting the rotate_on_start_only parameter to zero (0). **Note:** Log rotation does not occur until Microsoft Outlook is restarted.
reuse_not_exceeded	Determines the file that Oracle CRM On Demand Desktop uses when a new logging session starts. You can use one of the following values: - **1.** If the size of the most current log file is less than the value that the max_size_bytes parameter sets, then Oracle CRM On Demand Desktop reuses this file. The default value is one. - **0.** Oracle CRM On Demand Desktop does not reuse the most current log file.
rotate_on_start_only	Log file rotation is a configuration that Oracle CRM On Demand Desktop uses to limit the size of log files. You can use one of the following values: - **1.** Examines the size of the log file size when Outlook starts. If this log file is larger than the value that the max_size_bytes parameter specifies, then Oracle CRM On Demand Desktop creates a new log file, and then it logs all subsequent entries to this new file. The log file size can exceed the value that the max_size_bytes parameter specifies. Oracle CRM On Demand Desktop stores all log entries for one Outlook session in one file. The default value is 1. - **0.** Examines the size of the log file every time Oracle CRM On Demand Desktop writes to the log file. If this log file size is larger than the value that the max_size_bytes parameter specifies, then Oracle CRM On Demand Desktop creates a new log file. Multiple log files can exist for a single Outlook session.
file_count	An integer that specifies the maximum number of rotated log files.
initial_erase	You can use one of the following values: - **1.** Deletes the contents of all general log files when Oracle CRM On Demand Desktop starts.

Parameter	Description
	• **0.** Does not delete the contents of any general log file when Oracle CRM On Demand Desktop starts.
time_format	A string that specifies the date-and-time format in the log files. You can use one of the following values to specify the date-and-time format: • **$nano.** Nanoseconds. • **$micro.** Microseconds. • **$mili.** Milliseconds. • **$ss.** The seconds expressed as two digits. • **$mm.** The minutes expressed as two digits. • **$hh**. The hour expressed as two digits. • **$dd.** The day expressed as two digits. • **$MM.** The month expressed as two digits. • **$yy.** The year expressed as two digits. • **$yyyy.** he year expressed as four digits. Nanoseconds or microseconds are available only if the system clock allows nanoseconds or microseconds. If the system clock does not allow nanoseconds or microseconds, and if you specify nanoseconds or microseconds, then Oracle CRM On Demand Desktop adds zeros to the value that it creates for the log entry.
log_format	A string that specifies the logging message format. You can use one of the following values: • **%index%.** Sequential message index. • **%time%.** The time in the format that the time_format parameter sets. • **%thread%.** The thread ID that Oracle CRM On Demand Desktop uses to log the message. • **%level%.** The logging level for messages in human-readable format. • **%level_num%.** The logging level for messages in numeric format. • **%log_src%.** The source of the logging message. This source can be an internal component of Oracle CRM On Demand Desktop, for example, a connector or application manager.
starter	A string that specifies the first message that Oracle CRM On Demand Desktop adds to the log file when it starts logging, for example: --- logging is initialized ---
finisher	A string that specifies the last message that Oracle CRM On Demand Desktop adds to the log file when it stops logging, for example: --- logging is finished --- If a log file does not include the last message, then the log file indicates that Oracle CRM On Demand Desktop stopped logging abnormally.

Windows Registry Parameters That You Can Use with the Exception Log File

The following table describes the Windows parameters that you can use with the exception log file.

Parameter	Description
enable	You can use one of the following values: • **1.** The exception log file is enabled. • **0.** The exception log file is disabled.
file_count	An integer that specifies the maximum number of rotated log files.
initial_erase	You can use one of the following values: • **1.** Deletes the contents of all general log files when Oracle CRM On Demand Desktop starts. • **0.** Does not delete the contents of any general log file when Oracle CRM On Demand Desktop starts.
max_size_bytes	An integer that sets the maximum number of bytes that a single log file can contain.
out_file	A string that specifies the base file name that Oracle CRM On Demand Desktop uses for logging.

Windows Registry Parameters That You Can Use with the Crash Dump File

The following table describes the Windows parameters that you can use with the crash dump file. You can use the following parameters:

- any_in_trace
- all_in_trace
- ex_white_list
- ex_black_list

To separate the values in these parameters, you can use the following symbols:

- , (comma)
- ; (semicolon)
- | (vertical bar)
- / (slash)
- \t (backslash with a t)
- (single space)

Parameter	Description
enable	You can use one of the following values: - **1.** The general log is enabled. - **0.** The general log is disabled.
Count	An integer that specifies the maximum number of old dump files that Oracle CRM On Demand Desktop preserves in the output logging directory.
on_top	A string that specifies the filtering condition for the crash dump file. Oracle CRM On Demand Desktop applies this filtering only if the item you specify resides on top of the stack trace when the failure occurs. For example, you can use the following value: `CRMDesktop3D.dll` If you do not specify a value, then Oracle CRM On Demand Desktop ignores this parameter.
on_top_op	You can use one of the following values: - **0.** Combines the item that the on_top parameter filters with the item that the any_in_trace parameter specifies or with the item that the all_in_trace parameter specifies. Use the OR logical operator. - **1.** Combines the item that the on_top parameter filters with the item that the any_in_trace parameter specifies and with the item that the all_in_trace parameter specifies. Use the AND logical operator. - **2.** Combines the item that the on_top parameter does not filter with the item that the any_in_trace parameter specifies or with the item that the all_in_trace parameter specifies. Use the OR logical operator. - **3.** Combines the item that the on_top parameter does not filter with the item that the any_in_trace parameter specifies and with the item that the all_in_trace parameter specifies. Use the AND logical operator.
any_in_trace	A string that specifies a list of items. If the stack trace includes an item that you specify, then Oracle CRM On Demand Desktop saves a crash dump file for this item. For example, you can use the following value: `Ntdll.dll,Kernel32.dll` In this example, if the stack trace includes an entry for Ntdll.dll or Kernel32.dll, then Oracle CRM On Demand Desktop saves a crash dump file for this item.
all_in_trace	A string that specifies a list of items. If the stack trace includes all the items you specify, then Oracle CRM On Demand Desktop saves a crash dump file for these items. For example, you can use the following value: `Ntdll.dll,Kernel32.dll` In this example, if the stack trace includes an entry for Ntdll.dll and an entry for Kernel32.dll, then Oracle CRM On Demand Desktop saves a crash dump file for these items.
delay_ms	An integer that specifies a delay in milliseconds. If the time that elapses after a previous exception exceeds the value you specify, then Oracle CRM On Demand Desktop saves a crash dump file. The default value is 0. This default makes sure that Oracle CRM On Demand Desktop always saves a crash dump file.

Parameter	Description
ex_white_list	A string that specifies a list of C++ exception names. If an exception occurs for an exception name that you specify, then Oracle CRM On Demand Desktop creates a minidump for this exception.
ex_black_list	A string that specifies a list of C++ exception names that Oracle CRM On Demand Desktop uses but does not create a minidump file. If the white_list parameter is empty, then Oracle CRM On Demand Desktop creates a dump file for all exceptions except the exceptions that the black_list parameter lists.
sigabrt	SIGABRT is a signal that Oracle CRM On Demand Desktop sends to a process to tell it to end. You can use one of the following values: • **1.** Intercepts the SIGABRT signal to save a crash dump file. You must use this parameter only for troubleshooting. Do not enable it permanently. • **0.** Does not intercept the SIGABRT signal to save a crash dump file.
minidump_type	An integer that specifies the mindump type. The default value is 1 (MiniDumpNormal| MiniDumpWithDataSegs). Use this parameter with caution because other values might increase the size of the crash dump file significantly. For more information, see the documentation about the DumpType parameter of the MiniDumpWriteDump function at the Microsoft MSDN Web site.

Windows Registry Parameters That You Can Use with the SOAP Dump File

The following table describes the Windows parameters that you can use with the SOAP dump file.

Parameter	Description
enable	You can use one of the following values: • **1.** The SOAP dump is enabled. • **0.** The SOAP dump is disabled.
enable_dbg_window	You can use one of the following values: • **1.** Enables output through the OutputDebugString. The SOAP dump displays in the VS Debug Output window during a debugging session. • **0.** Disables output through the OutputDebugString. The SOAP dump does not display in the VS Debug Output window during a debugging session.
enable_cout	You can use one of the following values: • **1.** Enables logging to the console. • **0.** Disables logging to the console.
file_count	An integer that specifies the maximum number of rotated log files.

Parameter	Description
root_tag_name	A string that specifies the root XML tags that Oracle CRM On Demand Desktop uses in generated XML files. The default value is soap_comm_log. If you use the following value, then it does not write the opening or closing root tags: ""
file_extension	A string that specifies the file extension that Oracle CRM On Demand Desktop uses for generated log files. The default value is xml.
bin_mode	You can use one of the following values: • **1.** Oracle CRM On Demand Desktop uses the \n format to terminate a new line. • **0.** Oracle CRM On Demand Desktop uses the \r\n format to terminate a new line.

Windows Registry Parameters That You Can Use with the Synchronization Dump File

The following table describes the Windows parameters that you can use with the synchronization dump file.

Parameter	Description
enable	You can use one of the following values: • **1.** The synchronization log is enabled. • **0.** The synchronization log is disabled.
file_count	An integer that specifies the maximum number of rotated log files.
out_file	A string that specifies the base file name that Oracle CRM On Demand Desktop uses for logging.

15 Code Listing for Updates to the Customization Package from Version 2.0

Code Listing for Updates to the Customization Package from Version 2.0

This chapter lists some of the code updates required when updating the customization package from Oracle CRM On Demand Desktop Version 2.0. The appendix includes the following topics:

- *XML Code Addition for the od_meta_info.xml File*
- *JavaScript Code Addition for the od_helpers.js File*
- *XML Code Addition for the od_basic_mapping.xml File*
- *JavaScript Code Addition for the forms.js File*

XML Code Addition for the od_meta_info.xml File

Insert the following XML code inside the `<root>` element in the od_meta_info.xml file, as described in Step k in the topic *Setting Up Books for a Customized Package from Version 2.0*:

```
<object TypeId='BookRaw' Label='BookRaw' LabelPlural='Books' ViewMode='Sales Rep'
IntObjName='Book' XmlElemName='Book' XmlCollectionElemName='ListOfBook'
DeleteIsProhibited="yes">
 <direct_ws_params DataNS="urn:/crmondemand/xml/Book/Query"
InputElemname="BookQueryPage_Input" InputNS="urn:crmondemand/ws/ecbs/book/"
OutputElemname="BookQueryPage_Output" OutputNS="urn:crmondemand/ws/ecbs/book/"
SOAPAction="document/urn:crmondemand/ws/ecbs/book/:BookQueryPage"
WrapQBEVals="false" PagingSupported="true" />
 <field Name="BookName" Label="BookName" DataType="DTYPE_TEXT" CRMName="Name" />
 <field Name="BookType" Label="BookType" DataType="DTYPE_TEXT" CRMName="Book
Type" />
 <field Name="CanContainDataFlag" Label="CanContainDataFlag"
DataType="DTYPE_BOOL" CRMName="Can Contain Data Flag" />
 <field Name="Description" Label="Description" DataType="DTYPE_TEXT" />
 <field Name="Id" Label="Id" DataType="DTYPE_TEXT" IsPrimaryKey="yes"
IsFilterable='no' IsHidden="yes" />
 <field Name="ModId" Label="ModId" DataType="DTYPE_INTEGER" IsTimestamp="yes"
IsFilterable='no' IsHidden="yes" />
 <field Name="ParentBookId" Label="ParentBookId" DataType="DTYPE_TEXT"
IsFilterable='no' />
 <field Name="ParentBookName" Label="ParentBookName" DataType="DTYPE_TEXT"
CRMName="Parent Book Name" />
</object>

<!--
This type draws its data from BookPL picklist, which in turn is based on Book type.
It was introduced because currently platform caches picklists, so we will not see
any picklists updates while OL instance lives.
Book type could be used directly instead, but that would lead to duplicated Book
downloads (picklist is always loaded and then we would have to execute query_changes/
query_objects/... calls for that same type).
Existence if this type allows to just take existing data from the connector picklists
cache.
-->
<object TypeId='Book' Label='Book' LabelPlural='BookPLBased' >
 <svc_params Type="PicklistBased" Path="[Book]@[::Defaults]" />
 <field Name="Id" Label="Id" DataType="DTYPE_ID" IsCalculated="true"
Formula=":[:(Value):]" IsPrimaryKey="yes" IsFilterable='no' IsHidden="yes"/>
 <field Name="BookId" Label="BookId" DataType="DTYPE_TEXT" IsCalculated="true"
Formula=":[:(Value):]" IsFilterable='no'/>
 <field Name="BookName" Label="BookName" DataType="DTYPE_TEXT"
IsCalculated="true" Formula=":[:(Label):]" IsFilterable='no'/>
 <field Name="Value" Label="Value" DataType="DTYPE_TEXT" />
 <field Name="Label" Label="Label" DataType="DTYPE_TEXT" />
 <field Name="ParentBookName" Label="ParentBookName" DataType="DTYPE_TEXT" />
 <field Name="ParentBookId" Label="ParentBookId" DataType="DTYPE_TEXT" />
 <field Name="BookType" Label="BookType" DataType="DTYPE_TEXT" />
 <field Name="CanContainDataFlag" Label="CanContainDataFlag"
DataType="DTYPE_BOOL" />
 <field Name="OrderNumber" Label="OrderNumber" DataType="DTYPE_TEXT" />
</object>
<object TypeId='BookDefaults' Label='BookDefaults' LabelPlural='BookDefaults' >
 <svc_params Type="PicklistBased" Path="[TypeDefaultBook]@[::Defaults]" />
 <field Name="Id" Label="Id" DataType="DTYPE_ID" IsCalculated="true"
Formula=":[:(Value):]" IsPrimaryKey="yes" IsFilterable='no' IsHidden="yes"/>
 <field Name="Value" Label="Value" DataType="DTYPE_TEXT" />
 <field Name="TypeName" Label="TypeName" DataType="DTYPE_ID" IsCalculated="true"
Formula=":[:(Value):]" IsFilterable='no'/>
 <field Name="Label" Label="Label" DataType="DTYPE_TEXT" />
 <field Name="BookId" Label="BookId" DataType="DTYPE_TEXT" />
 <field Name="IncludeSubBooks" Label="IncludeSubBooks" DataType="DTYPE_BOOL" />
 <field Name="OrderNumber" Label="OrderNumber" DataType="DTYPE_TEXT" />
</object>
<object TypeId="Book.Type.Association" Label="Book.Type.Association"
LabelPlural="Book.Type.Associations" >
 <svc_params Type="SettingsBased" Path="RemoteConnectorStorage/
:[:(storage_id):]/Types" />
 <field Name="BookId" Label="BookId" DataType="DTYPE_TEXT" />
 <field Name="TypeName" Label="TypeName" DataType="DTYPE_TEXT" />
 <field Name="IncludeSubbooks" Label="IncludeSubBooks" DataType="DTYPE_BOOL" />
</object>
<object TypeId='RequiredFields' Label='RequiredFields'
LabelPlural='RequiredFields'>
 <svc_params Type="PicklistBased" Path="[RequiredFields]@[::Defaults]" />
 <field Name="Id" Label="Id" DataType="DTYPE_ID" IsCalculated="true"
Formula=":[:(Value):]@:[:(TypeName):]" IsPrimaryKey="yes" IsFilterable='no'
IsHidden="yes"/>
 <field Name="FieldName" Label="FieldName" DataType="DTYPE_TEXT"
IsCalculated="true" Formula=":[:(Value):]" IsFilterable='no'/>
 <field Name="Value" Label="Value" DataType="DTYPE_TEXT" />
 <field Name="Label" Label="Label" DataType="DTYPE_TEXT" />
 <field Name="TypeName" Label="TypeName" DataType="DTYPE_TEXT" />
 <field Name="Required" Label="Required" DataType="DTYPE_BOOL" />
 <field Name="OrderNumber" Label="OrderNumber" DataType="DTYPE_TEXT" />
</object>

<object TypeId='CustomRecordType' Label='CustomRecordType'
LabelPlural='CustomRecordTypes' XmlElemName='CustomRecordType'
DeleteIsProhibited="yes" >
 <direct_ws_params DataNS="urn:/crmondemand/xml/customrecordtype/query"
InputElemname="CustomRecordTypeRead_Input" InputNS="urn:crmondemand/ws/odesabs/
customrecordtype/" OutputElemname="CustomRecordTypeRead_Output"
OutputNS="urn:crmondemand/ws/odesabs/customrecordtype/" SOAPAction="document/
urn:crmondemand/ws/odesabs/CustomRecordType/:CustomRecordTypeRead"
WrapQBEVals="false" SuppressListOfElem="request" PagingSupported="false" URL="/
Services/cte/CustomRecordTypeService" />
```

```
 <field Name="AccessObjName" Label="AccessObjName" DataType="DTYPE_TEXT" />
 <field Name="Name" Label="Name" DataType="DTYPE_TEXT" IsPrimaryKey = "yes" />
 <field Name="IncludeAll" Label="IncludeAll" DataType="DTYPE_BOOL"
FillOnQueryWith="true" />
</object>

<object TypeId='FieldSet' Label='FieldSet' LabelPlural='FieldSets' ViewMode='Sales
Rep' XmlElemName='FieldSet' DeleteIsProhibited="yes" >
 <direct_ws_params DataNS="urn:/crmondemand/xml/fieldmanagement/query"
InputElemname="FieldManagementRead_Input" InputNS="urn:crmondemand/ws/odesabs/
fieldmanagement/" OutputElemname="FieldManagementRead_Output"
OutputNS="urn:crmondemand/ws/odesabs/fieldmanagement/" SOAPAction="document/
urn:crmondemand/ws/odesabs/FieldManagement/:FieldManagementRead"
WrapQBEVals="false" SuppressListOfElem="request" PagingSupported="false" URL="/
Services/cte/FieldManagementService" />
 <field Name="ObjectName" Label="ObjectName" DataType="DTYPE_TEXT" IsPrimaryKey =
"yes" />
 <field Name="IncludeAll" Label="IncludeAll" DataType="DTYPE_BOOL" />
</object>

<object TypeId='FieldSet.Fields' Label='FieldSet.Fields'
LabelPlural='FieldSet.Fields' ViewMode='Sales Rep' XmlElemName='Field'
XmlCollectionElemName='ListOfFields' DeleteIsProhibited="yes" >
 <field Name="TypeName" Label="TypeName" ActualFldName="ObjectName"
ActualObjTypeId="FieldSet" IsRequired="yes" DataType="DTYPE_TEXT" IsNullable="no"
IsFilterable="no" IsRefObjId="yes" RefObjTypeId="FieldSet" RefObjIsParent="yes"
IsFake="yes" />
 <field Name="IncludeAll" Label="IncludeAll" ActualFldName="IncludeAll"
ActualObjTypeId="FieldSet" DataType="DTYPE_TEXT" IsFilterable="no" IsFake="yes" />
 <field Name="Name" Label="Name" DataType="DTYPE_TEXT" IsPrimaryKey = "yes" />
 <field Name="IntegrationTag" Label="IntegrationTag" DataType="DTYPE_TEXT"
IsPrimaryKey = "yes" />
 <field Name="GenericIntegrationTag" Label="GenericIntegrationTag"
DataType="DTYPE_TEXT" />
 <field Name="Required" Label="Required" DataType="DTYPE_BOOL" />
</object>

<object TypeId="User.DefaultBooks" Label="User.DefaultBooks"
LabelPlural="User.DefaultBooks" ViewMode="Sales Rep"
XmlElemName="UserDefaultBookPerType"
XmlCollectionElemName="ListOfUserDefaultBookPerType" DeleteIsProhibited="yes" >
 <field Name="AccessObjectId" Label="AccessObjectId" DataType="DTYPE_TEXT" />
 <field Name="AccessObjectName" Label="AccessObjectName" DataType="DTYPE_TEXT" />
 <field Name="BookDescription" Label="BookDescription" DataType="DTYPE_TEXT" />
 <field Name="BookId" Label="BookId" DataType="DTYPE_TEXT" IsRefObjId="yes"
RefObjTypeId="BookRaw" IsFilterable='no' />
 <field Name="BookName" Label="BookName" DataType="DTYPE_TEXT" />
 <field Name="Id" Label="Id" DataType="DTYPE_ID" IsPrimaryKey="yes"
IsFilterable='no' IsHidden="yes" />
 <field Name="IncludeSubBooks" Label="IncludeSubBooks" DataType="DTYPE_BOOL" />
 <field Name="ModId" Label="ModId" DataType="DTYPE_INTEGER" IsTimestamp="yes"
IsFilterable='no' IsHidden="yes" />
 <field Name="UserId" Label="UserId" ActualFldName="Id" ActualObjTypeId="User"
IsRequired="yes" DataType="DTYPE_ID" IsNullable="no" IsFilterable="no"
IsRefObjId="yes" RefObjTypeId="User" RefObjIsParent="yes" IsFake="yes" />
 <!-- inherited from parent -->
 <field Name="CurrencyCode" Label="Currency Code" DataType="DTYPE_TEXT"
ActualFldName="CurrencyCode" ActualObjTypeId="User" IsFilterable="no" />
 <!-- <field Name="DefaultBookId" Label="DefaultBookId"
ActualFldName="DeftBookId" ActualObjTypeId="User" DataType="DTYPE_TEXT"
IsFilterable="no" /> -->
 <field Name="DefaultBookName" Label="DefaultBookName"
ActualFldName="DefaultBookName" ActualObjTypeId="User" DataType="DTYPE_TEXT"
IsFilterable="no" />
 <field Name="FullName" Label="Full Name" DataType="DTYPE_TEXT" CRMName="Full
Name" ActualFldName="FullName" ActualObjTypeId="User" IsFilterable="no" />
```

```
 field Name="LanguageCode" Label="Language Code" ActualFldName="LanguageCode"
ActualObjTypeId="User" DataType="DTYPE_TEXT" IsFilterable="no" />
 <field Name="PrimaryPositionId" Label="PrimaryPositionId" DataType="DTYPE_TEXT"
ActualFldName="PrimaryPositionId" ActualObjTypeId="User" IsFilterable="no" />
 <field Name="RoleId" Label="Role Id" ActualFldName="RoleId"
ActualObjTypeId="User" DataType="DTYPE_TEXT" IsFilterable="no" />
 <field Name="SalesProcessId" Label="Sales Process Id"
ActualFldName="SalesProcessId" ActualObjTypeId="User" DataType="DTYPE_TEXT"
IsFilterable="no" />
 <field Name="SalesProcessName" Label="Sales Process Name"
ActualFldName="SalesProcessName" ActualObjTypeId="User" DataType="DTYPE_TEXT"
IsFilterable="no" />
 <field Name="UserSignInId" Label="User Sign In Id" DataType="DTYPE_TEXT"
IsHidden="yes" ActualFldName="UserSignInId" ActualObjTypeId="User" />
</object>

<picklist TypeId='TypeDefaultBookPL' SrcObjectTypeId='User.DefaultBooks'
ValueFldName='AccessObjectName' LabelFldName='AccessObjectName' Artificial="true">
 <extra_src_fldname Visible="true">BookId</extra_src_fldname>
 <extra_src_fldname Visible="true">IncludeSubBooks</extra_src_fldname>
</picklist>

<picklist TypeId='BookPL' SrcObjectTypeId='BookRaw' ValueFldName='Id'
LabelFldName='BookName' OrderByExpression=':[:(BookName):]'>
 <extra_src_fldname Visible="true">ParentBookName</extra_src_fldname>
 <extra_src_fldname Visible="true">ParentBookId</extra_src_fldname>
 <extra_src_fldname Visible="true">BookType</extra_src_fldname>
 <extra_src_fldname Visible="true">CanContainDataFlag</extra_src_fldname>
</picklist>

<picklist TypeId='FieldsInfoPL' SrcObjectTypeId='FieldSet.Fields'
ValueFldName='IntegrationTag' LabelFldName='IntegrationTag'
OrderByExpression=':[:(IntegrationTag):]'>
 <extra_src_fldname Visible="true">TypeName</extra_src_fldname>
 <extra_src_fldname Visible="true">Required</extra_src_fldname>
 <extra_src_fldname Visible="false" OutVal="true">IncludeAll<
 extra_src_fldname>
</picklist>
```

JavaScript Code Addition for the od_helpers.js File

Insert the following JavaScript code after the first line of the file, as described in Step 8 in the topic *Setting Up Books for a Customized Package from Version 2.0*:

```
include("form_helpers.js", "form_helpers");
var allow_includesubbooks =
{
// "Account": true
};
function book_selector(ctx, options)
{
 var session = ctx.session,
 ui = ctx.ui,
 dlg,
 tree,
 image = "",
 on_changed;

 var btn = options.ctrl_btn,
 edit = options.ctrl_edit,
 id_field = options.field || "BookId",
 ownership_mode = options.mode,
 context = options.context,
 in_tree = {};

 edit.enabled = false;
 on_changed = new helpers.signal();
 this.on_changed = on_changed;
 this.refresh = refresh;
 if(context == "inspector")
 {
 ctx.events.connect(btn, "on_click", show);
 dlg = ctx.ui.create_dialog(0, "select_book_dialog");
 tree = dlg.books_tree;
 ctx.events.connect(tree, "on_node_activated", select);
 ctx.events.connect(dlg.btn_cancel, "on_click", on_cancel);
 ctx.events.connect(dlg.btn_ok, "on_click", on_save);
 ctx.events.connect(dlg.btn_clear, "on_click", on_clear);
 dlg.btn_clear.visible = true;
 btn.enabled = ownership_mode != "owner";
 this.refresh();
 }

 this.set_visible = function(visible)
 {
 edit.visible = btn.visible = visible;
 }

 function fill_tree(parent_book_id, parent_node, seelected_book_id)
 {
 parent_book_id = parent_book_id || "";
 parent_node = parent_node || tree;
 var add_node = function(item)
 {
 var book_id = item.Value
 book_name = item.Label;
 image = "book_" + (item.CanContainDataFlag ? "can" : "cannot") +
"_contain_data_icon";
 var node = parent_node.nodes.add(book_name, image, {"id": book_id});
 if(book_id == seelected_book_id && book_id)
 tree.selected = node;
 in_tree[book_id] = true;
 fill_tree(book_id, node, seelected_book_id);
 }
 for_each_child(parent_book_id, add_node);
 // handle orphans
 if(parent_book_id == "") // top level
 {
 helpers.for_each(session.find_items("Book", session.empty_criteria),
function(item)
 {
 var has_no_parent = session.find_item("Book",
session.create_expression("Value", "eq", item.ParentBookId)) == null;
 if(!in_tree[item.Value] && has_no_parent)
 add_node(item);
 });
 }
 };
 function show()
 {
 dlg.visible = true;
 tree.nodes.remove_all();
 in_tree = {};
 fill_tree(null, null, ctx.item_ex.get_property(id_field));
 // Tree refresh workaround
 dlg.btn_clear.set_focus();
 tree.set_focus();
 }

 function refresh()
 {
 edit.value = id_to_label(ctx.item_ex.get_property(id_field));
 }

 function on_cancel(){dlg.close();}

 function on_save(){
 var selected_node = tree.selected;
 select(selected_node);
 }


 function on_clear(){
 tree.selected = null;
 }

 function select(node)
 {
 edit.value = node != null ? node.name : "";
 var book_id = node != null ? node.data.id : "";
 ctx.item_ex.set_property(id_field, book_id);
 dlg.close();
 on_changed.raise(book_id); }


 function for_each_child(parent_book_id, fn)
 {
 helpers.for_each(session.find_items("Book",
session.create_expression("ParentBookId", "eq", parent_book_id)), fn);
 }

 function id_to_label(id)
 {
 var book = id != null ? session.find_item("Book",
session.create_expression("Value", "eq", id)) : null;
 var label = "";
 if(book != null)
 label = book.Label;
 else
 if(id != null && id != "")
 {
 var user = ctx.session.find_item("User",
ctx.session.create_expression("PrimaryPositionId", "eq", id));
 if(user != null)
 label = user["FullName"];
 else
 label = session.res_string("inaccessible_book");
 }
 return label;
 }
 function get_current_user_userbook(ctx)
 {
 var defaults = helpers.get_defaults(ctx.session);
 return defaults != null ? defaults.PrimaryPositionId : "";
 }}
function book_can_contain_data(ctx, id)

{
 var book = ctx.session.find_item("Book", ctx.session.create_expression("Value",
"eq", id)),
 result = true;
 if(book != null)
 result = book.CanContainDataFlag;
 return result;
}
function book_is_userbook(ctx, id)
{
 var user = ctx.session.find_item("User",
ctx.session.create_expression("PrimaryPositionId", "eq", id));
 return user != null;
}
function get_ownership_mode(ctx, type)
{
 if(type == "Event" || type == "Task" || type == "Mail")
 type = "Activity";
 var mode = "owner";
 var books_enabled = ctx.session.find_item("Book",
ctx.session.create_criteria("or")) != null;
 if(books_enabled)
 {
 var owner_filter = ctx.session.create_criteria("and");
 owner_filter.add(ctx.session.create_expression("TypeName", "eq", type));
 owner_filter.add(ctx.session.create_expression("Required", "eq", true));
 owner_filter.add(ctx.session.create_expression("Value", "eq", "Owner"));
 var owner_required = ctx.session.find_item("RequiredFields", owner_filter) !=
null;
 if(!owner_required) {
 var book_filter = ctx.session.create_criteria("and");
 book_filter.add(ctx.session.create_expression("TypeName", "eq", type));
 book_filter.add(ctx.session.create_expression("Value", "eq",
"BookName"));
 var oField = ctx.session.find_item("RequiredFields", book_filter);
 var book_required = oField != null ? oField.Required : null;
 if (book_required)
 mode = "book";
 else
 mode = "mixed";
 }
 }
 return mode;
}
function cp_book_selector(ctx, options)
{
 var dlg;
 var tree;
 var image = "";
 var btn = options.ctrl_btn;
 var edit = options.ctrl_edit;
 var books = options.books;
 var allow_include_subbooks = options.allow_include_subbooks;
 var selected_book;
 var on_changed = new helpers.signal();
 this.on_changed = on_changed;

 edit.enabled = false;
 ctx.events.connect(btn, "on_click", show);
 dlg = ctx.ui.create_dialog(0, "select_book_dialog");
 dlg.include_subitems.visible = true;
 dlg.include_subitems.enabled = allow_include_subbooks;
 tree = dlg.books_tree;
 ctx.events.connect(tree, "on_node_activated", select);
 ctx.events.connect(tree, "on_node_selected", uncheck_include_subitems);
 ctx.events.connect(dlg.btn_cancel, "on_click", on_cancel);
 ctx.events.connect(dlg.btn_ok, "on_click", on_save);

 this.set_visible = function(visible) {
 edit.visible = btn.visible = visible;
 };
 this.enable = function(enabled) {
 btn.enabled = enabled;
 }
 this.update_options = function (new_options) {
 btn = new_options.ctrl_btn || btn;
 edit = new_options.ctrl_edit || edit;
 books = new_options.books || books;
 };
 function show() {
 dlg.visible = true;
 tree.nodes.remove_all();
 fill_tree();
 // Tree refresh workaround
 dlg.include_subitems.set_focus();
 tree.set_focus();
 }

 function on_cancel() { dlg.close(); }

 function on_save() {
 select(tree.selected);
 }

 function select(node) {
 var selected_book_id = node != null ? node.data.id : null;
 var selected_include_subitems = node != null ? dlg.include_subitems.checked :
null;
 if (selected_book_id) {
 if (selected_include_subitems ||
books[selected_book_id].CanContainDataFlag) {
 set_book_selected({BookId: selected_book_id, IncludeSubBooks:
selected_include_subitems});
 uncheck_include_subitems();
 dlg.close();
 } else {
 var message = (allow_include_subbooks ?
"msg_subitems_required_for_book" : "msg_cannot_select_book_subitems_disallowed");
 ctx.ui.message_box(0, ctx.session.res_string(message),
ctx.session.res_string("msg_validation_failed"), 0x30);
 }
 } else {
 dlg.close();
 }
 }

 var set_book_selected = function (book) {
 selected_book = book;
 refresh(book);
 on_changed.raise(book);
 }
 this.set_book_selected = set_book_selected;

 function uncheck_include_subitems() { dlg.include_subitems.checked = false; }

 function fill_tree(parent_book_id, parent_node) {
 parent_book_id = parent_book_id || "";
 parent_node = parent_node || tree;
 for_each_child(parent_book_id, add_node);
 // handle orphans
 if (parent_book_id == "") {
 helpers.for_each2(books, function(item) {
 var parent_inaccessible = books[item.ParentBookId] == null;
 if (item.ParentBookId != "" && parent_inaccessible)
 add_node(item);
 });
 }
 function add_node(item) {
 var book_id = item.Value,
 book_name = item.Label;
 image = "book_" + (item.CanContainDataFlag ? "can" : "cannot") +
"_contain_data_icon";
 var node = parent_node.nodes.add(book_name, image, {"id": book_id});
 if(selected_book && book_id && selected_book.BookId == book_id) {
 tree.selected = node;
 dlg.include_subitems.checked = dlg.include_subitems.visible &&
!!selected_book.IncludeSubBooks;
 }
 fill_tree(book_id, node);
 }
 function for_each_child(parent_book_id, fn) {
 var child_books = [];
 helpers.for_each2(books, function (item) {
 if (item.ParentBookId == parent_book_id)
 child_books.push(item);
 });
 helpers.for_each2(child_books, fn);
 }
 };
 function refresh(book) {
 // Update book label
 var book_id = book != null ? book.BookId : null;
 var label = "";
 if (book_id != null && book_id != "")
 label = books[book_id] != null
 ? (books[book_id].Label + (book.IncludeSubBooks ? "+" : ""))
 : ctx.session.res_string("inaccessible_book");
 edit.value = label;
 }
}
function record_set_page(ctx)
{
 var parent_form, form = ctx.form;
 var cached_bs_controllers = {};
 var saved_item_snapshot, current_item;
 var record_set_types = [];
 var book_processor, predefined_options = {};
 ctx.events.connect(form.btn_save1, "on_click", on_save);

 ctx.events.connect(form, "on_action_requested", on_action_requested);

 function saved_books_retrieved(retrieved_books) {
 current_item = retrieved_books;
 rebuild_saved_item_snapshot();

 // Switch back to active page UI from on_busy
 switch_back_from_daisy();
 form.head_record_type.visible = form.head_visibility_model.visible =
form.head_book_name.visible = true;

 // Get list of supported types
 record_set_types = book_processor.get_types_list();

 // Clear previous records set controls table
 var i = 0;
 helpers.for_each2(record_set_types, function(){
 var img_type = form["img_type" + i],
 lbl_type = form["lbl_type" + i],
 cbx_mode = form["cbx_mode" + i],
 edit_book = form["edit_book" + i],
 btn_book_select = form["btn_book_select" + i];

 img_type.image = null;
 lbl_type.caption = "";
 cbx_mode.value = null;
 img_type.visible = lbl_type.visible = cbx_mode.visible = edit_book.visible
= btn_book_select.visible = false;
 i++;
 });

 build_table();
 }
 function build_table() {
 var row_num = 0;
 // Disable "Save" button initially
 validate_changes();
 // Build table
 helpers.for_each2(record_set_types, function(rs_type){
 var img_type = form["img_type" + row_num],
 lbl_type = form["lbl_type" + row_num],
 cbx_mode = form["cbx_mode" + row_num],
 edit_book = form["edit_book" + row_num],
 btn_book_select = form["btn_book_select" + row_num];
 var book_available = rs_type.books_enabled &&
book_processor.user_has_books();

 // Make type image, caption and mode selector visible
 var caption_lbl = rs_type.type_id;
 try { img_type.image = "type_image:"+rs_type.type_id+":24"; } catch(e){}
 try { caption_lbl =
ctx.session.res_string("obj_"+rs_type.type_id.toLowerCase()); } catch(e){}
 lbl_type.caption = caption_lbl;
 img_type.visible = lbl_type.visible = cbx_mode.visible = true;
 cbx_mode.enabled = edit_book.visible = btn_book_select.visible =
book_available;
 var current_type_settings;
 book_processor.apply_to_type(rs_type.type_id,
function(item){current_type_settings = item}, current_item); // set
current_type_settings
 // Build mode selector options
 cbx_mode.items.clear();
 cbx_mode.items.start_adding();
 cbx_mode.items.add("team",
ctx.session.res_string("cbx_record_set_team"));
 if (book_available || current_type_settings.BookId != null)
 cbx_mode.items.add("book",
ctx.session.res_string("cbx_record_set_book"));
 if (book_available && book_processor.default_book_exists(rs_type) ||
is_default_book_selected(current_type_settings, rs_type.default_book))
 cbx_mode.items.add("default_book",
ctx.session.res_string("cbx_record_set_default_book"));
 cbx_mode.items.finish_adding();

 // Retrieve cached book selector (or create a new one if not found)
 var books = book_processor.get_available_books();
 var bs = cached_bs_controllers[rs_type.type_id];
 if (bs) {
 bs.update_options({type: rs_type, books: books});
 } else {
 var bs_options =
 {
 ctrl_edit: edit_book,
 ctrl_btn: btn_book_select,
 type: rs_type,
 books: books,
 allow_include_subbooks:
book_processor.allow_include_subbooks(rs_type.type_id)
 };
 bs = new cp_book_selector(ctx, bs_options);
 ctx.events.connect(bs, "on_changed", set_option_selected);
 cached_bs_controllers[rs_type.type_id] = bs;
 ctx.events.connect(cbx_mode, "changed",
function(){on_mode_changed(cbx_mode.value)});
 }

 // Fill controls with initial values
 prefill_type(current_type_settings);
 function set_option_selected(book) {
 book_processor.apply_to_type(rs_type.type_id, function(item){
 item.BookId = book.BookId;
 item.IncludeSubBooks = book.IncludeSubBooks;
 }, current_item);
 validate_changes();
 }

 function on_mode_changed(mode) {
 bs.set_visible(mode !== "team");
 switch (mode) {
 case "team":
 set_option_selected(predefined_options.team);
 break;
 case "book":
 bs.enable(true);
 if (book_processor.default_book_exists(rs_type))
 bs.set_book_selected(rs_type.default_book);
 else
 bs.set_book_selected(predefined_options.no_book_selected);
 break;
 case "default_book":
 bs.enable(false);
 bs.set_book_selected(rs_type.default_book);
 break;
 }
 }

 /**
 Prefills mode selector and appropriate controls with passed item values.
 @param item Item of type Book.Type.Association
 @type Function
 */
 function prefill_type(item) {
 var book_id = item.BookId;
 var include_subbooks = item.IncludeSubBooks;
 var mode = "team";
 if (book_id !== predefined_options.team.BookId)
 if (is_default_book_selected(item, rs_type.default_book))
 mode = "default_book";
 else
 mode = "book";
 cbx_mode.value = mode;
 if(mode != "team")
 cbx_mode.enabled = true; // user should be able to switch to "team"
mode in any case
 on_mode_changed(mode);
 if (mode == "book")
 bs.set_book_selected({"BookId": book_id, "IncludeSubBooks":
include_subbooks});
 }

 function is_default_book_selected(type_item, default_book_item) {
 return type_item.BookId == default_book_item.BookId &&
type_item.IncludeSubBooks == default_book_item.IncludeSubBooks
 }

 row_num++;
 });
 }

 function on_save() {
 book_processor.save(current_item, on_saved);
 function on_saved(){
 rebuild_saved_item_snapshot();
 validate_changes();
 }
 }
 function on_action_requested(ar) {
 switch (ar.action) {
 case "set_parent":
 parent_form = ar.parent_form;
 break;
 case "page_select":
 on_page_select();
 break;
 case "query_close": // CP closing
 if (form.btn_save1.enabled)
 {
 var res = ctx.application.ui.message_box(parent_form,
ctx.session.res_string("msg_record_set_query_save_record_set"),
ctx.session.res_string("lbl_record_set_caption"), 35);
 ar.cancel = res == 2;
 if (res == 6 && book_processor != null)
 on_save();
 }
 break;
 case "close": // CP closed
 on_close();
 break;
 }
 }

 function on_page_select() {
 var bp_options = {
 "connector": book_processor ? book_processor.get_connector() : null
 };
 book_processor = new book_of_business_processor(ctx, bp_options);
 predefined_options = book_processor.get_predefined_options();
 parent_form.infobar.caption =
ctx.session.res_string("record_set_tab_description");
 parent_form.infobar.visible = parent_form.toolbar.visible = false;
 parent_form.cp_pages.active_page = parent_form.cp_pages.pages.cp_busy_page;
 book_processor.save_default_books_if_empty(saved_books_retrieved,
on_connector_failed);
 }

 function on_close() {
 if (book_processor)
 book_processor.abort();
 }
 function on_connector_failed() {
 // close Control Panel
 parent_form.close();
 }

 /**
 Switches back to active page UI from on_busy page with daisy.
 @type Function
 */
 function switch_back_from_daisy() {
 parent_form.cp_pages.active_page =
parent_form.cp_pages.pages.cp_record_set_page;
 parent_form.infobar.visible = parent_form.toolbar.visible = true;
 }
 /**
 Enables/disables "Save" button based on detected changes and performed
validations.
 @type Function
 */
 function validate_changes() {
 var enable_save_btn = true;
 var changes_found = false;
 // Detect changes
 if (saved_item_snapshot.length != current_item.length) {
 changes_found = true;
 } else {
 for (var i in current_item) {
 var current_item_options = current_item[i];
 var saved_item_snapshot_options = saved_item_snapshot[i];
 if (saved_item_snapshot_options.BookId != current_item_options.BookId
|| saved_item_snapshot_options.IncludeSubBooks !=
current_item_options.IncludeSubBooks) {
 changes_found = true;
 break;
 }
 }
 }

 if (!changes_found) {
 enable_save_btn = false;
 } else {
 // Run validations
 for (var i in current_item) {
 if (current_item[i].BookId ===
predefined_options.no_book_selected.BookId) {
 enable_save_btn = false;
 break;
 }
 }
 }
 // Enable/disable "Save" button
 form.btn_save1.enabled = enable_save_btn;
 }
 /**
 Builds a snapshot of current_item, which can later be used as saved item snapshot
for detecting changes.
 @type Function
 */
 function rebuild_saved_item_snapshot() {
 var data_source = book_processor.get_data_source();
 var columns = data_source.columns.toArray();
 saved_item_snapshot = [];
 for (var key in current_item) {
 var snapshot_item = {};
 for (var i in columns)
 snapshot_item[columns[i]] = current_item[key][columns[i]];
 saved_item_snapshot.push(snapshot_item);
 }
 }
}

function book_of_business_processor(ctx, options) {
 options = options || {};
 var available_books, user_default_book, user_has_books;
 var record_set_types = [];
 var default_books = [];
 var data_source, connector = options.connector || null;
 var disconnectable = {};
 var book_predefined_options =
 {
 "team": {BookId: "", IncludeSubBooks: false},
 "no_book_selected": {BookId: null, IncludeSubBooks: null}
 };

 this.save_default_books_if_empty = function(on_after_saved_fn,
on_save_failed_fn)

{
 var on_books_retrieved = function (retrieved_books) {
 save(retrieved_books, on_after_saved_fn);
 };
 this.retrieve_saved_or_default_books(on_books_retrieved, on_save_failed_fn);
 };

 this.retrieve_saved_or_default_books = function(on_books_retrieved_fn,
on_err_fn)
{
 // Saving default visibility modes
 if (!connector)
 ctx.application.connector.async_get_remote_connector(0,
form_helpers.disconnectable_handler(on_connector_ready, disconnectable));
 else
 on_connector_ready(connector);

 function on_connector_ready(c, err) {
 if (err) {
 ctx.ui.message_box(ctx.ui.app_window, err.message,
ctx.session.res_string("books_retrieve_failed_title"), 0x10); // MB_OK |
MB_ICONERROR
 if (on_err_fn)
 on_err_fn(err);
 return;
 }

 // Reset variables
 var saved_item = [];
 var default_prefilling = [];
 record_set_types = [];
 default_books = [];
 available_books = [];
 user_default_book = null;
 data_source = null;

 // Cache connector
 connector = c;
 // Retrieve remotely available books list and default books per type for
current user
 create_data_source("Book", function(book_ds) {
 available_books = retrieve_books_picklist_from_datasource(book_ds);
 var books_count = 0;
 for (var _ in available_books) { books_count++; }
 user_has_books = books_count > 0;
 create_data_source("BookDefaults", function(BookDefaults_ds) {
 default_books =
retrieve_books_picklist_from_datasource(BookDefaults_ds);
 on_books_ready();
 });
 });

 function on_books_ready()
 {
 // Configure list of supported types
 for (var type_id in default_books)
 record_set_types.push({type_id: type_id, sort_order:
parseInt(default_books[type_id]["OrderNumber"])});
```

```
 record_set_types.sort(function(a, b){ return a.sort_order -
b.sort_order; });

 // Get global default book
 create_data_source("::Defaults", function(defaults_ds) {
 // If no items found, proceed
 if (defaults_ds.record_count == 0) {
 configure_default_books();
 } else {
 var ds_item = defaults_ds.get_item_id(0);
 connector.async_get_item("::Defaults", ds_item, function(){},
form_helpers.disconnectable_handler(function(defaults, err) {
 if (err == null && defaults.DefaultBookName != "" &&
defaults.DefaultBookName != "All+") {
 for (var id in available_books) {
 if (available_books[id].Label ==
defaults.DefaultBookName) {
 user_default_book = id;
 break;
 }
 }
 }
 configure_default_books();
 }, disconnectable));
 }

 function configure_default_books() {
 helpers.for_each2(record_set_types, function(rs_type){
 // If default book per type is not available set to global
default book if available
 if (default_books[rs_type.type_id].BookId == null &&
user_default_book != null) {
 default_books[rs_type.type_id].BookId = user_default_book;
 default_books[rs_type.type_id].IncludeSubBooks =
!available_books[user_default_book].CanContainDataFlag &&
allow_include_subbooks(rs_type.type_id);
 }
 // Check if books of business are enabled
 rs_type.books_enabled = default_books[rs_type.type_id] !=
null;
 // Lookup default book for type if books enabled
 if (rs_type.books_enabled)
 rs_type.default_book = default_books[rs_type.type_id];
 // Default prefilling
 var book_available = rs_type.books_enabled && user_has_books;
 var def_book_exists = default_book_exists(rs_type);
 var prefill_default_book = book_available && def_book_exists;
 apply_to_type(rs_type.type_id, function(item){
 item.BookId = prefill_default_book ?
rs_type.default_book.BookId : book_predefined_options.team.BookId;
 item.IncludeSubBooks = prefill_default_book ?
rs_type.default_book.IncludeSubBooks :
book_predefined_options.team.IncludeSubBooks;
 }, default_prefilling);
 });
 // Get saved item if found
 create_data_source("Book.Type.Association", function(ds) {
 // Make datasource global for future use
 data_source = ds;
 // If no saved items found, consider this is a first run and
proceed next
 if (data_source.record_count > 0) {
 // Process found items
 var items_processed = 0;
 for (var i=0; i<data_source.record_count; i++) {
 var ds_item = data_source.get_item_id(i);
 connector.async_get_item("Book.Type.Association",
ds_item, function(){},
form_helpers.disconnectable_handler(function(retrieved_item, err) {
 if (err == null)
 saved_item.push(retrieved_item);
 items_processed++;
 // If all items retrieved
 if (items_processed == data_source.record_count)
 configure_default_item();
 }, disconnectable));
 }
 } else {
 configure_default_item();
 }

 // Compiles default visibility modes list
 function configure_default_item() {
 helpers.for_each2(record_set_types, function(rs_type){
 // Trying to find saved mode
 var item_found = false;
 for (var key in saved_item) {
 if (saved_item[key].TypeName == rs_type.type_id) {
 item_found = true;
 break;
 }
 }
 // set to default mode if not found
 if (!item_found)
 apply_to_type(rs_type.type_id, function(item){
 for (var key in default_prefilling) {
 if (default_prefilling[key].TypeName ==
item.TypeName) {
 item.BookId =
default_prefilling[key].BookId;
 item.IncludeSubBooks =
default_prefilling[key].IncludeSubBooks;
 break;
 }
 }
 }, saved_item);
 });

 on_books_retrieved_fn(saved_item);
 }
 });
 }
 });
 }

 function retrieve_books_picklist_from_datasource(ds) {
 var result = {};
 var count = ds.record_count;
 var _columns = ds.columns.toArray();
 var columns = {};
 for (var i=0; i < _columns.length; i++) {
 columns[_columns[i]] = i;
 }
 for (var i=0; i < count; i++) {
 var row = ds.get_item_fields(i).toArray();
 result[get_value(row)] = get_item(row);
 }

 function get_value(row)
 {
 return row[columns["Value"]];
 }
 function get_item(row)
 {
 var item = {};
 helpers.for_each2(_columns, function(fld_name){
 item[fld_name] = row[columns[fld_name]];
 });
 return item;
 }
 return result;
 }

 function create_data_source(type_id, fn) {
 var ds = connector.create_data_source(type_id);
 ds.on_changed.connect(form_helpers.disconnectable_handler(function() {
fn(ds); }, disconnectable));
 ds.enabled = true;
 ds.refresh(100, 600000);
 ds.enabled = false;
 }
 }
 }

 function save(item, on_after_saved_fn) {
 var saved_items_count = 0;
 for (var key in item) {
 connector.async_put_item(item[key], on_progress,
form_helpers.disconnectable_handler(on_put_item_complete, disconnectable));
 }
 if (item.length == 0)
 on_after_saved_fn(item);

 function on_progress() {}
 function on_put_item_complete(saved_book) {
 // Synchronize new selected value with local storage
 var user_selected_book = get_ol_storage_rs_book(ctx, saved_book.TypeName)
|| ctx.session.create_item("Book.Type.Association");
 user_selected_book.TypeName = saved_book.TypeName;
 user_selected_book.BookId = saved_book.BookId;
 user_selected_book.IncludeSubBooks = saved_book.IncludeSubBooks;
 user_selected_book.save();

 saved_items_count++;
 if (saved_items_count == item.length)
 on_after_saved_fn(item);
 }
 }
 this.save = save;

 this.get_types_list = function () {
 return record_set_types;
 }
 this.get_data_source = function () {
 return data_source;
 }

 this.get_connector = function () {
 return connector;
 };

 this.get_available_books = function () {
 return available_books;
 };

 this.user_has_books = function () {
 return user_has_books;
 }
 this.get_predefined_options = function () {
 return book_predefined_options;
 }

 this.abort = function() {
 disconnectable.disabled = true;
 }

 function allow_include_subbooks(type) {
 return allow_includesubbooks[type] || false;
 }
 this.allow_include_subbooks = allow_include_subbooks;

 /**
 Checks whether default book is defined for a type.
 @param type Type name to check default book for
 @returns {Boolean} Does default book exist for a type
 @type Function
 */
 function default_book_exists(type) {
 // Books are disabled
 if (!type.books_enabled)
 return false;
 // No default book
 var book_id = type.default_book.BookId;
 if (book_id == null)
 return false;
 // accessible book && the book that can contain data or with "Include Sub-
Items" enabled by administrator
 var book = available_books[book_id];
 if (book != null && (book.CanContainDataFlag ||
type.default_book.IncludeSubBooks))
 return true;
 // The book that cannot contain data and "Include Sub-Items" for it is disabled
by administrator
 return false;
 }
 this.default_book_exists = default_book_exists;

 /**
 Applies a function to a specific type records set option in destination item. if
not yet exists, creates "team" records set option for a type (default option).
 @param type_id Records set option type name
 @param {Function} f Function to call over an item
 @param dest Destination object to lookup an item in.
 @type Function
 */
 function apply_to_type(type_id, f, dest) {
 // Try to find item in dest
 var item = null;
 for (var i in dest) {
 if (dest[i].TypeName == type_id) {
 item = dest[i];
 break;
 }
 }
 // Create item if not found
 if (item == null) {
 item = connector.create_item("Book.Type.Association");
 item.TypeName = type_id;
 item.BookId = book_predefined_options.team.BookId;
 item.IncludeSubBooks = book_predefined_options.team.IncludeSubBooks;
 dest.push(item);
 }
 // Apply function
 f(item);
 }
 this.apply_to_type = apply_to_type;
}

function get_ol_storage_rs_book(ctx, type_id) {
 return ctx.session.find_item("Book.Type.Association",
ctx.session.create_expression("TypeName", "eq", type_id));
}
```

XML Code Addition for the od_basic_mapping.xml File

Add new types by inserting the following lines inside the `<database> <types>` element in the od_basic_mapping.xml as described in Step c in the topic *Setting Up Books for a Customized Package from Version 2.0*:

```
<type id = "Book" icon = "type_image:Generic:16">
 <field id = "Label">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "Value">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "SortOrder">
 <type>
 <simple type = "integer"/>
 </type>
 </field>
 <field id = "IsDefault">
 <type>
 <simple type = "boolean"/>
 </type>
 </field>
 <field id = "ParentCode">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "ParentBookName">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "ParentBookId">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "BookType">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "CanContainDataFlag">
 <type>
 <simple type = "boolean"/>
 </type>
 </field>
```

```
     </type>
    <type id = "BookDefaults" icon = "type_image:Generic:16">
     <field id = "Label">
     <type>
     <simple type = "string"/>
     </type>
     </field>
     <field id = "Value">
     <type>
     <simple type = "string"/>
     </type>
     </field>
     <field id = "SortOrder">
     <type>
     <simple type = "integer"/>
     </type>
     </field>
     <field id = "IsDefault">
     <type>
     <simple type = "boolean"/>
     </type>
     </field>
     <field id = "ParentCode">
     <type>
     <simple type = "string"/>
     </type>
     </field>
     <field id = "BookId">
     <type>
     <simple type = "string"/>
     </type>
     </field>
     <field id = "IncludeSubBooks">
     <type>
     <simple type = "boolean"/>
     </type>
     </field>
    </type>
    <type id = "RequiredFields" icon = "type_image:Generic:16">
     <field id = "Label">
     <type>
     <simple type = "string"/>
     </type>
     </field>
     <field id = "Value">
     <type>
     <simple type = "string"/>
     </type>
     </field>
     <field id = "SortOrder">
     <type>
     <simple type = "integer"/>
     </type>
     </field>
     <field id = "IsDefault">
     <type>
     <simple type = "boolean"/>
     </type>
     </field>
     <field id = "ParentCode">
     <type>
     <simple type = "string"/>
     </type>
     </field>
     <field id = "TypeName">
     <type>
```

```
 <simple type = "string"/>
 </type>
 </field>
 <field id = "Required">
 <type>
 <simple type = "boolean"/>
 </type>
 </field>
 <field id = "SalesProcLang">
 <type>
 <simple type = "string"/>
 </type>
 </field>
</type>
<type id = "Book.Type.Association">
 <field id = "TypeName">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "BookId">
 <type>
 <simple type = "string"/>
 </type>
 </field>
 <field id = "IncludeSubBooks">
 <type>
 <simple type = "boolean"/>
 </type>
 </field>
</type>
```

JavaScript Code Addition for the forms.js File

Insert the register_book_control(ctx, owner_ctrl) function at the end of the file, as described in Step a in the topic *Setting Up Books for a Customized Package from Version 2.0*:

```
function register_book_control(ctx, owner_ctrl)
{
 var ownership_mode = ctx.security_descriptor.ownership_mode(),
 form = ctx.form,
 type = form.item.type_id,
 current_user_link = { "item_ex": ctx.item_ex, "with_id":
helpers.get_current_user_id(ctx.session), "tag": "direct" },
 current_user_link_mvg = helpers.merge_contexts(current_user_link, { "tag":
"mvg", "params": {"is_manual_operation": false} });
 if(!is_activity_subform(type))
 {
 ctx.validator.add_custom(util_validate_book_fab(ctx, ownership_mode),
["book_of_business"], "msg_book_is_required");
 if(owner_ctrl)
 {
 ctx.validator.add_custom(util_validate_owner_fab(ownership_mode),
[owner_ctrl], "msg_owner_must_be_empty");
 ctx.validator.add_custom(util_validate_owner_and_book_fab(ctx,
ownership_mode), [owner_ctrl, "book_of_business"],
"msg_Specifying_both_Book_and_Owner_is_prohibited");
 }
 }
 var bs = new od_helpers.book_selector(ctx, {ctrl_edit: form.book_of_business,
ctrl_btn: form.btn_book_select, mode: ownership_mode, context: "inspector"});
```

```
 if(owner_ctrl)
 ctx.triggers.add_simple_trigger(on_owner_changed, ctx.item_ex.get_type(),
"User", "direct", "created");
 ctx.events.connect(form, "on_saving", function() {
 // clean-up fields
 switch(ownership_mode)
 {
 case "owner":
 if(ctx.item_ex.get_property("OwnerId") == null)
 ctx.linker.link(current_user_link);
 ctx.linker.link(current_user_link_mvg);
 book_clean_up();
 break;
 case "book":
 ctx.item_ex.set_property("OwnerId", null);
// ctx.linker.unlink(current_user_link_mvg);
 break;
 }
 });
 ctx.events.connect(bs, "on_changed", function(book_id) {
 // populate/clean-up Owner
 if(ownership_mode == "mixed")
 if(book_id != "")
 {
 ctx.item_ex.set_property("OwnerId", null);
 ctx.linker.unlink(current_user_link_mvg);
 }
 else
 {
 if(!owner_ctrl) // no link to User
 ctx.item_ex.set_property("OwnerId", current_user_link.with_id);
 else
 {
 ctx.linker.link(current_user_link);
 ctx.linker.link(current_user_link_mvg);
 }
 }
 });
 function is_activity_subform(type)
 {
 return type == "Event" || type == "Task" || type == "Mail";
 }
 function on_owner_changed(ctx, action_ctx)
 {
 var owner_id = action_ctx.with_id;
 if(ownership_mode == "mixed" && owner_id != null)
 book_clean_up();
 }
 function book_clean_up()
 {
 ctx.item_ex.set_property("BookId", "");
 bs.refresh();
 }
 function util_validate_book_fab(ctx, ownership_mode)
 {
 return function(validation_ctx)
 {
 var book = validation_ctx.snapshot()["BookId"];
 if(ownership_mode == "book" && (book == null || book == ""))
 {
 return false;
 }
 else if(ownership_mode != "owner" &&
!od_helpers.book_can_contain_data(ctx, book))
 {
 validation_ctx.set_message("msg_book_cannot_contain_data", true);
 return false;
 }
 else if(ownership_mode == "book" && od_helpers.book_is_userbook(ctx,
book))
 {
 validation_ctx.set_message("msg_book_must_be_custom", true);
 return false;
 }
 else
 return true;
 }
 }
 function util_validate_owner_fab(ownership_mode)
 {
 return function(validation_ctx)
 {
 var owner = validation_ctx.snapshot()["OwnerId"];
 return !(ownership_mode == "book" && owner != null)
 }
 }
 function util_validate_owner_and_book_fab(ctx, ownership_mode)
 {
 return function(validation_ctx)
 {
 var owner = validation_ctx.snapshot()["OwnerId"];
 if(ownership_mode == "mixed")
 {
 var book = validation_ctx.snapshot()["BookId"];
 if(owner != null && book != "" && book != null &&
!od_helpers.book_is_userbook(ctx, book))
 return false;
 }
 return true;
 }
 }}
```

16 Add-In Compatibility in Oracle CRM On Demand Desktop

Add-In Compatibility in Oracle CRM On Demand Desktop

This chapter provides a list of add-ins that are compatible and incompatible with Oracle CRM On Demand Desktop. It is recommended that you refer to this appendix before installing any add-ins with Oracle CRM On Demand Desktop. The appendix includes the following topics:

- *Add-Ins Compatible with Oracle CRM On Demand Desktop*
- *Add-Ins Incompatible with Oracle CRM On Demand Desktop*

Add-Ins Compatible with Oracle CRM On Demand Desktop

The following add-ins are compatible with Oracle CRM On Demand Desktop:

- OneNote Notes
- Redemption Helper Outlook Extension
- Apple iTunes
- Cisco WebEx
- Skype Toolbar
- Outlook Social Connector
- LinkedIn
- Mindjet MindManager
- GoodApp
- Microsoft Outlook Hotmail Connector
- Lync Meeting add-in for Microsoft Office

Add-Ins Incompatible with Oracle CRM On Demand Desktop

The following add-ins are incompatible with Oracle CRM On Demand Desktop:

- Evernote Clipper for Outlook

- G Suite Sync for Microsoft Outlook (Google Apps Sync for Microsoft Outlook)
- Microsoft Business Contact Manager
- Microsoft Office Live Meeting
- Microsoft Access Outlook Add-in for Data Collection and Publishing
- Xobni
- Taskline
- IronPort
- EMC SourceOne
- Colligo.SendPlusFile
- TeamViewer for Meetings
- Microsoft Dynamics CRM

17 Glossary

access control

The set of Oracle CRM On Demand mechanisms that control the records to which the user possesses access and which operations the user can perform on the records.

account

A financial entity that represents the relationships between a company and the companies and people with whom the company does business.

account team

Users who possess access to the account record. A user who is assigned to the account is a member of the account team.

Active X

A specific way to implement ActiveX technology. It denotes reusable software components that use the component object model (COM) from Microsoft. ActiveX controls provide functionality that is encapsulated and reusable in programs. They are typically, but not always, visual in nature.

Active X control

A specific way to implement ActiveX technology. It denotes reusable software components that use the component object model (COM) from Microsoft. ActiveX controls provide functionality that is encapsulated and reusable in programs. They are typically, but not always, visual in nature.

activity

A task that a user must track. Examples include a to-do item, email sent to a contact, or an appointment with a contact

activity (Oracle CRM On Demand)

An object in Oracle CRM On Demand that organizes, tracks, and resolves a variety of work, from finding and pursuing an opportunity to closing a service request. An activity also captures an event, such as scheduling a meeting or appointment that occurs at a specific time and is displayed in the calendar.

appointment (Microsoft Outlook)

A record in the Microsoft Outlook calendar or Oracle CRM On Demand application calendar that reserves time to perform something, such as an appointment to schedule a meeting with a customer or to reserve time to complete work in a given period.

attendee (Microsoft Outlook)

A person included in the appointment, such as an organizer or a participant.

authentication

The process of verifying the identity of a user.

business object

A logical representation of Oracle CRM On Demand entities, such as accounts, opportunities, activities, and contacts, and the logical groupings and relationships among these entities. A business object uses links to group business components into logical units. The links provide the one-to-many relationships that govern how the business components interrelate in this business object. For example, the opportunity business object groups the opportunity, contact, and activities business components.

business object (activity)

The object that is the parent of or related to the activity, for example, a service request, opportunity, marketing campaign, order orchestration process, and so on.

business object (interaction)

The object that is the focus of the communication between the customer and the organization. For example, a service request, opportunity, contract, and so on.

child business component

A business component that represents the many in the one-to-many relationship between two business components in a parent-child relationship.

child record

An instance of the child business component.

contact

A person with whom a user might be required to phone or email to pursue a selling relationship. Various business objects can refer to a contact, but this reference does not require a relationship between the customer and contact. In Oracle CRM On Demand, a contact attribute in the context of a business object is a party that might or might not have a relationship defined. In Microsoft Outlook, a contact attribute in the context of a business object is the same as the Contact folder. Therefore, a contact can be a consumer and can also be a user of an organization.

contact points

Methods of contacting a contact other than through a postal address, such as email, telephone, and fax.

ConversationId

An attribute on an appointment record in Microsoft Outlook that the user can use to correlate shared appointments between meeting attendees. Meeting attendees in Microsoft Outlook include their own copy of the appointment, but all copies include the same value for the ConversationId.

CRM (Customer Relationship Management)

A software application that helps a business track customer interactions.

CRM contact

A contact whose details are recorded within a CRM application.

current view

The Microsoft Outlook view that displays content from the Microsoft Outlook folder that is currently chosen

custom view

A view that a user creates to control the amount of detail that displays in a particular folder. The user can create a filter or change the order of the columns and how the columns are arranged in the new custom view.

customer

A party with whom a user maintains a selling relationship. This party can be an organization or a person. Various business objects can refer to a customer. In Oracle CRM On Demand, a customer attribute in the context of a business object can be a person or an organization that includes the party usage type of Customer. In Microsoft Outlook, a customer attribute in the context of a business object can be an organization or a contact that is flagged as a consumer.

customer team

A group of users from the deploying organization or partners who actively work with a customer, including nonsales personnel, such as product marketing, partners, or customer service.

customization

The process of changing the definition of Oracle CRM On Demand Desktop.

customization package

A logical collection of metadata files that are associated with a particular responsibility. A customization package is released to the client computer.

data synchronization

The process of checking for differences between two or more different sets of data, then updating the data sets so that the data in each set is consistent.

form

A generic concept that Microsoft Outlook uses to present information about a single record and data related to that record in a form layout. Each control in the form is a separate attribute or collection of related data. A form can also support different tabs so that details of a child record can be displayed as separate lists.

hash value

A fixed-size string that is obtained as a result of cryptographic transformation from a cryptographic hash function.

homepage

A user interface component in Microsoft Outlook that displays a collection of information from Microsoft Outlook and CRM applications, and potentially external Web content that is embedded.

household

Provides a way to group consumers.

inbound Web service

A Web service that Oracle CRM On Demand makes available.

incremental synchronization

An incremental synchronization is a synchronization session that occurs any time after the initial synchronization.

installation package

An installation executable that includes the application binaries and any necessary instructions for completing the customization package installation in Microsoft Outlook. It also includes details that are required to connect the application server for the initial synchronization.

interaction (Oracle CRM On Demand)

The tracking of customer communications with an organization in the context of the channels through which that communication occurs and the business objects to which they refer. An interaction can take the form of a phone call, email, chat request, Web collaboration, or communication through another channel.

interaction (Oracle CRM On Demand Desktop)

Provides a historical view of the communication that occurred. For example, the Sales team uses interactions to capture communications with a customer during the sales cycle.

interactive authentication

A type of authentication in Oracle CRM On Demand Desktop SSO that displays a separate Oracle CRM On Demand Desktop SSO Login dialog box in the client.

lead

An unqualified sales opportunity that often represents the first contact in the opportunity management process. After a lead is qualified it can be converted to an opportunity.

list view

A generic concept in a PIM application that presents information in a list. Each row in the list is a separate record and each column in the list is a separate field in the record.

logging profile

A logging profile is a set of parameters that determine logging settings.

lookup control

A control that is available in Microsoft Outlook that allows the user to view records in a list, then choose one or more records to associate with the current item. To identify a subset of data from which to choose the data, a lookup control typically includes the capability to specify a search condition.

meeting

An appointment in Microsoft Outlook that includes at least one participant.

metadata files

XML files that hold information on how the user experience must be shaped. Oracle CRM On Demand Desktop uses metadata files to perform field mapping with the user interface, lookup controls in the user interface, application object mapping, and general representation of the user interface.

Microsoft Outlook data

Data that is created in the native Microsoft Outlook application.

Microsoft Outlook folder

A folder in Microsoft Outlook that contains a collection of data, such as email messages in the Inbox folder, or sent email messages in the Sent Items folder. In the context of this book, a Microsoft Outlook folder might also contain Oracle CRM On Demand data.

Microsoft Outlook object

An entity that is native to Microsoft Outlook. Examples of Microsoft Outlook objects include an email, appointment, contact, and so on.

Microsoft Outlook add-in

A program that performs important work, including storing and displaying Oracle CRM On Demand data in native Microsoft Outlook and synchronizing PIM and non-PIM data with the Oracle CRM On Demand server. See also*PIM* , *Oracle CRM On Demand server*.

Microsoft Outlook portlet

A portlet that uses data in a Microsoft Outlook folder that includes a custom view filter. The Microsoft Outlook portlet includes ActiveX characteristics.

Microsoft Outlook standard view

A default Microsoft Outlook view that exists without Oracle CRM On Demand Desktop. A Microsoft Outlook view provides different ways of viewing the same information in a folder by placing the information in different arrangements and formats.

noninteractive authentication

A type of authentication in Oracle CRM On Demand Desktop SSO that reuses the login name and password from the Oracle CRM On Demand Desktop Login dialog box.

offline

A mode in which the user uses Oracle CRM On Demand Desktop but does not possess access to the Oracle CRM On Demand server. When in offline mode, Oracle CRM On Demand Desktop uses data in the local data to perform operations. Synchronization is delayed until the user is online.

online

A mode in which the user uses Oracle CRM On Demand Desktop while connected to the Oracle CRM On Demand server and synchronizes data with the Oracle CRM On Demand server at regular intervals, or when the user performs an update. Similar to offline mode, when in online mode Oracle CRM On Demand Desktop uses data in the local data store to perform operations.

online lookup

An Oracle CRM On Demand Desktop feature that allows you to search and view all records visible to you in the Oracle CRM On Demand Desktop database while you are connected to your network.

opportunity

A qualified sales engagement that represents potential revenue where a sales representative is willing to officially commit to the pipeline and to include revenue in the sales forecast. The sales representative monitors the opportunity life cycle. This representative might be compensated depending on the results of cumulative sales and potentially on how well the representative maintains details about the opportunity.

organization team

Includes the sales groups who possess ownership of the associated prospect, customer, or products with the opportunity. It can also include those who are working on a certain size of deal or with a specific sales stage and the partner organizations that can help to close the deal.

organizer

In Microsoft Outlook, the person who created the appointment.

Oracle CRM On Demand

Oracle's customer relationship management (CRM) solution that hosts CRM data and provides CRM functionality in a hosted Oracle CRM On Demand environment.

Oracle CRM On Demand data

Business data that is created in the Oracle CRM On Demand Desktop, data that is created in Oracle CRM On Demand, or data that resides on the Oracle CRM On Demand server. Examples include an opportunity, account, or activity.

Oracle CRM On Demand Desktop

A solution provided by Oracle that includes modifications to the standard Microsoft Outlook capabilities that allows the user to work with Oracle CRM On Demand records and business processes.

Oracle CRM On Demand server

The server that runs Oracle CRM On Demand. The Oracle CRM On Demand server processes business logic and data access for Microsoft Outlook.

participant

In native Microsoft Outlook, the person who is invited to the meeting.

participant of interaction

The people who participate in an interaction. The participant can include an internal representative of the organization, such as a resource, agent, sales representative, and so on. The participant can also include an external representative, such as a customer, contact of a customer, account, or a site. In a help desk or in a user self-service application, a participant can be a user.

personalization

The process where the user tailors the user interface and behavior of Microsoft Outlook.

PIM

Personal Information Manager, an application that typically helps a user to manage a list of contacts, calendar entries, email, and so on. Microsoft Outlook, Google email, and Thunderbird are examples of PIMs.

personal information manager (PIM)

See *PIM*.

PIM data

Personal information that refers to data that is stored in native Microsoft Outlook that relates to contacts, appointments, and so on.

Pin Mode for Records

An Oracle CRM On Demand Desktop feature that enables you to temporarily keep synchronized Oracle CRM On Demand Desktop records in Microsoft Outlook.

portlet

A user interface component that is managed and displayed in the home page. The home page is composed of multiple portlets.

recipient

The person who receives an email.

record

A specific instance of the business component, also known as an Oracle CRM On Demand record, or an object in native Microsoft Outlook, also known as a Microsoft Outlook record.

role

An entity in Oracle CRM On Demand that determines which views the user can access in Microsoft Outlook. For example, the role of the sales representative allows the user to access the My Opportunities view, whereas the role of the Oracle CRM On Demand application developer allows the user to access administration views. An Oracle CRM On Demand application developer or system administrator defines the responsibilities.

sales team

The users who possess access to an opportunity record. A user who creates the opportunity record is automatically part of the sales team. Other users can also be assigned to the sales team so that they can collaborate on the opportunity.

side pane

A user interface component that is available in native Microsoft Outlook, which is analogous to a task pane or action pane in Oracle CRM On Demand. This region of the user interface is typically available on the right side of the user interface. It displays a collection of data and actions that are appropriate for the context in which the user accesses data. The user can interact with this data.

Simple Object Access Protocol (SOAP)

See *SOAP*.

single sign-on

An authentication process that permits you to enter one name and password in order to access multiple applications.

SOAP

Simple Object Access Protocol, a protocol that allows a user or program to interact with Web services by exchanging XML messages that conform to SOAP.

standard Microsoft Outlook

The native Microsoft Outlook application without Oracle CRM On Demand Desktop.

synchronization

A process that exchanges transactions between Oracle CRM On Demand and Microsoft Outlook. This synchronization makes sure that Oracle CRM On Demand data is the same on the Oracle CRM On Demand server and in Microsoft Outlook.

synchronization filter

Criteria that are considered during data synchronization so that certain records are included and other records are excluded from processing during synchronization.

Task (Microsoft Outlook)

A part of a set of actions that accomplish a job, solve a problem, or complete an assignment. In the native Microsoft Outlook application, a task is a collection of simple business objects at the user level. A task can be used as a reminder and also as a tracking tool the effort involved.

Web services

Self-contained, modular applications that can be described, published, located, and called over a network. Web services perform encapsulated business functions, ranging from a simple request-reply to full business process interactions. Web services combine development that uses components and Internet standards and protocols, which include HTTP, XML code, and SOAP.

Web services description language (WSDL)

The XML formatted language that is used to describe a Web service. See also *SOAP*.