Oracle Fusion Cloud Customer Experience

Integrating Sales with Oracle E-Business Suite, Release 12.1.3

June 2023

Oracle Fusion Cloud Customer Experience
Integrating Sales with Oracle E-Business Suite, Release 12.1.3

June 2023

F77863-02

Author: David Yetter

Contents

Get Help i

1 About This Guide 1
Audience and Scope 1
Related Guides 1

2 Introduction to the Integration 3
Overview of Oracle E-Business Suite and Oracle CX Sales Integration: Overview 3
Integration Component Architecture 3
Integration Services 4
Integration Process Flows 5
Customer Data Matching with Oracle E-Business Suite 5
Contact Data Matching with Oracle E-Business Suite 6
Quote Creation in Oracle E-Business Suite 8
Configuration Roadmap 10

3 Oracle E-Business Suite Configuration 13
Software Requirements for Oracle E-Business Suite 13
Applying Oracle E-Business Suite Patches: Explained 13
Create a Generic Error Page 14
Assigning Roles to the Oracle E-Business Suite User: Explained 14
Expose Web Services 14
Deploy Web Services 15
Create a Quotes Menu 16

4 Oracle CX Sales Configuration 19
Before You Configure Oracle CX Sales 19
Software Requirements for Oracle CX Sales 19
Create References to Oracle E-Business Suite Web Service 19
Create and Activate Sandboxes 22

Publish Sandboxes 23
Overview of Functions 24
Overview of Global Functions 24
Creating Global Functions: Explained 25
Creating Object Functions: Explained 26
Create Validation Rules 26
Create Custom Objects 27
Create the Integration Configuration Custom Object 28
Create the XREF Custom Object 30
Create a Field Mapping Custom Object 32
Create the EBS Match Child Object 34
Configuring Standard Objects: Explained 38
Configure the Account Standard Object 39
Configure the EBS Match Child Object 42
Configure the Opportunity Standard Object 43
Connect Oracle CX Sales to Oracle E-Business Suite 46

5 **Postconfiguration Tasks** **49**

Map Products 49
Extracting the List of Oracle E-Business Suite Products: Explained 49
Extract a List of Oracle CX Sales Products 50
Map Products Manually 50
Enter Mappings into Oracle CX Sales 51
Map Product Groups 52
Guidelines for Hiding Oracle E-Business Suite UI Elements for Streamlined Navigation 55
Hide Oracle E-Business Suite UI Elements 60
Test the Integration 62
Configure the Integration with the Security Console 62

6 **Required Files** **65**

Required HTML File: Explained 65
Required Groovy Files 65

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Use help icons to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons.

Get Support

You can get support at *My Oracle Support*. For accessible support, visit *Oracle Accessibility Learning and Support*.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program*. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

1 About This Guide

Audience and Scope

This guide is intended for anyone who is involved in integrating Oracle E-Business Suite with Oracle CX Sales.

You must perform the integration steps in this guide to integrate Oracle E-Business Suite with Oracle CX Sales.

If you want to set up and work with the additional features of Oracle CX Sales, see Oracle CX Sales documentation on Oracle Help Center at https://docs.oracle.com.

This document describes features available to users under Oracle CX Sales, Oracle Fusion Service, and Oracle Engagement Cloud licensing agreements.

Related Guides

The following table lists related guides which provide more information about the integration tasks covered in this guide.

Title of Guide	Guide Description
Getting Started with Your Sales Implementation	Describes how to set up a sales automation solution in Oracle CX Sales using a case study to describe concepts and procedures.
Implementing CX Sales	Describes how to configure and set up CX Sales.
Oracle Applications Cloud Configuring Applications Using Application Composer	Describes how to use tools to extend and configure your applications.
Understanding Import and Export Management for CX Sales and Fusion Service	Describes how to import legacy and other data into Oracle CX Sales and Fusion Service using the Import and Export Management feature.
Oracle E-Business Suite User Guide	Describes core features of Oracle E-Business Suite.

Related Topics

2 Introduction to the Integration

Overview of Oracle E-Business Suite and Oracle CX Sales Integration: Overview

This guide outlines the implementation and configuration steps that are required to integrate customer- and opportunity-management processes in Oracle CX Sales with quote-management processes in Oracle E-Business Suite.

The integration is designed to support customers who want to take advantage of the latest capabilities of the Oracle CX Sales application, and use their existing investment in an on-premise enterprise resource planning (ERP) system that provides complex product configuration, pricing, or discount information. In the integration, Oracle E-Business Suite quotes can be created from Oracle CX Sales opportunities. The Oracle E-Business Suite quote UI is embedded within a tab in the Opportunity Detail screen in Oracle CX Sales.

This guide is designed to be used as a template. The guide is a starting point that shows how Oracle CX Sales and Oracle E-Business Suite can be connected to create a value-added business process and user experience. You must enter the documented configurations and install the documented patches to create the integration.

However, it isn't a turnkey solution. Each implementation of Oracle CX Sales and Oracle E-Business Suite is unique, and each customer has different needs that have led them to implement application configurations that support their unique business requirements. While the steps in this guide describe how to connect a non-configured Oracle CX Sales instance to a non-configured Oracle E-Business Suite instance, they can be combined with configurations that have already been applied to each instance.

Note that most settings related to security are not included in this document. For information and recommendations, see Oracle CX Sales Security Reference and Securing Oracle CX Sales both on Oracle Help Center (https://docs.oracle.com/en/).

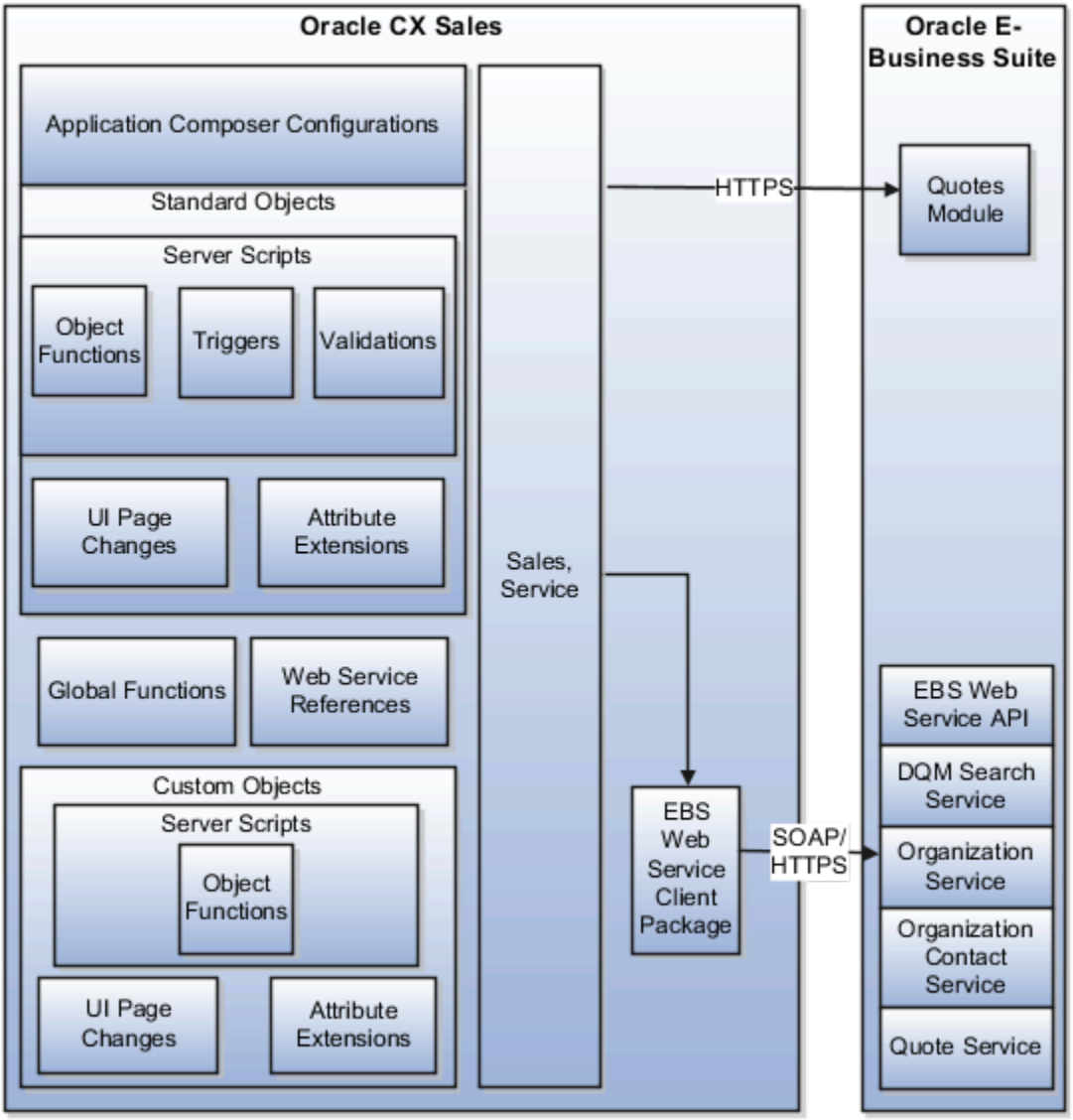
Integration Component Architecture

The following lists the files required to implement the integration between Oracle CX Sales and Oracle E-Business Suite.

Refer to the Related Topics section for a link to integration implementation files (Doc ID 1645923.1) on My Oracle Support. The files are located in the Attachments section of the article.

- **Oracle CX Sales standard objects**. Standard objects in the Sales, and Common applications are configured by adding fields, object functions, triggers, and validations, and by configuring UI pages.
- **Oracle CX Sales custom objects**. Custom objects are created for the integration, and fields, pages, functions, and so on are added to the objects.
- **Groovy functions**. Groovy scripts create functions, triggers, and validations that modify the behavior of Oracle CX Sales objects.
- **Oracle E-Business Suite web services**. Oracle CX Sales calls Oracle E-Business Suite web services directly. For more information on web services, see the Integration Services topic.

- **Oracle E-Business Suite Quotes module**. Opportunities in the Sales application in Oracle CX Sales are integrated with the Quotes module in Oracle E-Business Suite.



Related Topics

- Integration Services

Integration Services

The following Oracle E-Business Suite web services are used in the integration:

- **DQM Search Service.** This web service allows the user to search for organizations. It returns a list of candidates based on configurable criteria defined in a Data Quality Management (DQM) matching rule within Oracle Trading Community. DQM Search Service returns records that match the specified criteria. It returns duplicate and multiple records.
- **Organization Service.** This web service manages organization records in Oracle E-Business Suite. New organizations are created and retrieved by Oracle CX Sales.
- **Organization Contact Service.** This web service manages contact records for specific organizations in Oracle E-Business Suite. It creates new contacts in Oracle E-Business Suite from Oracle CX Sales.
- **Quote Service.** This web service can be used to create quotes in Oracle E-Business Suite based on opportunities in Oracle CX Sales.

Integration Process Flows

The integration between Oracle CX Sales and Oracle E-Business Suite supports the following three main process flows:

- Matching customer data between Oracle CX Sales and Oracle E-Business Suite
- Matching contact data with Oracle E-Business Suite
- Creating quotes in Oracle E-Business Suite

Customer Data Matching with Oracle E-Business Suite

This process lets you match customer records in Oracle CX Sales with Oracle E-Business Suite.

During the customer synchronization process, customer records must be searched in Oracle E-Business Suite before synchronizing. The process calls the DQM Search Service web service in Oracle E-Business Suite, which returns a list of potential matches. DQM Search Service also offers fuzzy-matching capabilities.

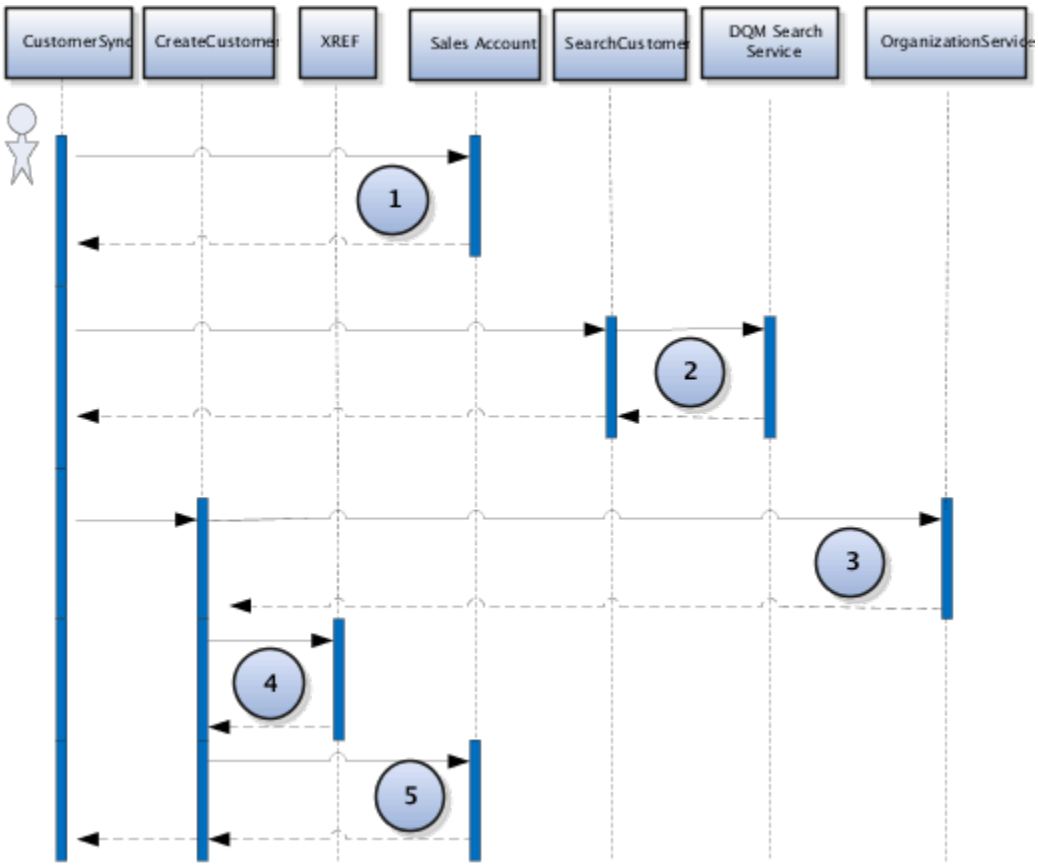
If the customer record exists in Oracle E-Business Suite and there is only one match, then the process saves a cross-reference in Oracle CX Sales as an XREF custom object. The cross-reference contains the local identification in Oracle CX Sales and the remote identification in Oracle E-Business Suite.

If the customer record exists in Oracle E-Business Suite and has multiple potential matches, then the user is presented with a user interface (UI) where the potential matches are displayed. The user selects the candidate match. The process then associates the record in Oracle E-Business Suite with the record in Oracle CX Sales by saving cross-references.

If the customer record does not exist in Oracle E-Business Suite, then the Organization Service web service is called in Oracle E-Business Suite to create a new customer record. The process then saves a cross-reference in Oracle CX Sales with the local identification in Oracle CX Sales and the remote identification in Oracle E-Business Suite.

Detailed Customer Data Matching Process

The following figure illustrates the customer data matching sequence.



When a salesperson performs a customer matching operation in Oracle CX Sales the following sequence takes place:

1. The process returns the Account object from Oracle CX Sales, and determines the customer synchronization status.
2. If the customer is not synchronized, or if the status is null, then the process calls the FindParties method in the DQM Search Service web service to perform a customer search in Oracle E-Business Suite. The customer search returns a list of matching candidates.
3. If there is no matching record, then the process calls the CreateOrganization method in the Organization Service web service in Oracle E-Business Suite. Organization Service returns the remote customer identification.
4. The process creates an XREF custom object in Oracle CX Sales using the local identification in Oracle CX Sales and the remote identification from Oracle E-Business Suite.
5. The process sets the synchronization status to Synchronized.

Contact Data Matching with Oracle E-Business Suite

This process lets you match contact records in Oracle CX Sales with Oracle E-Business Suite.

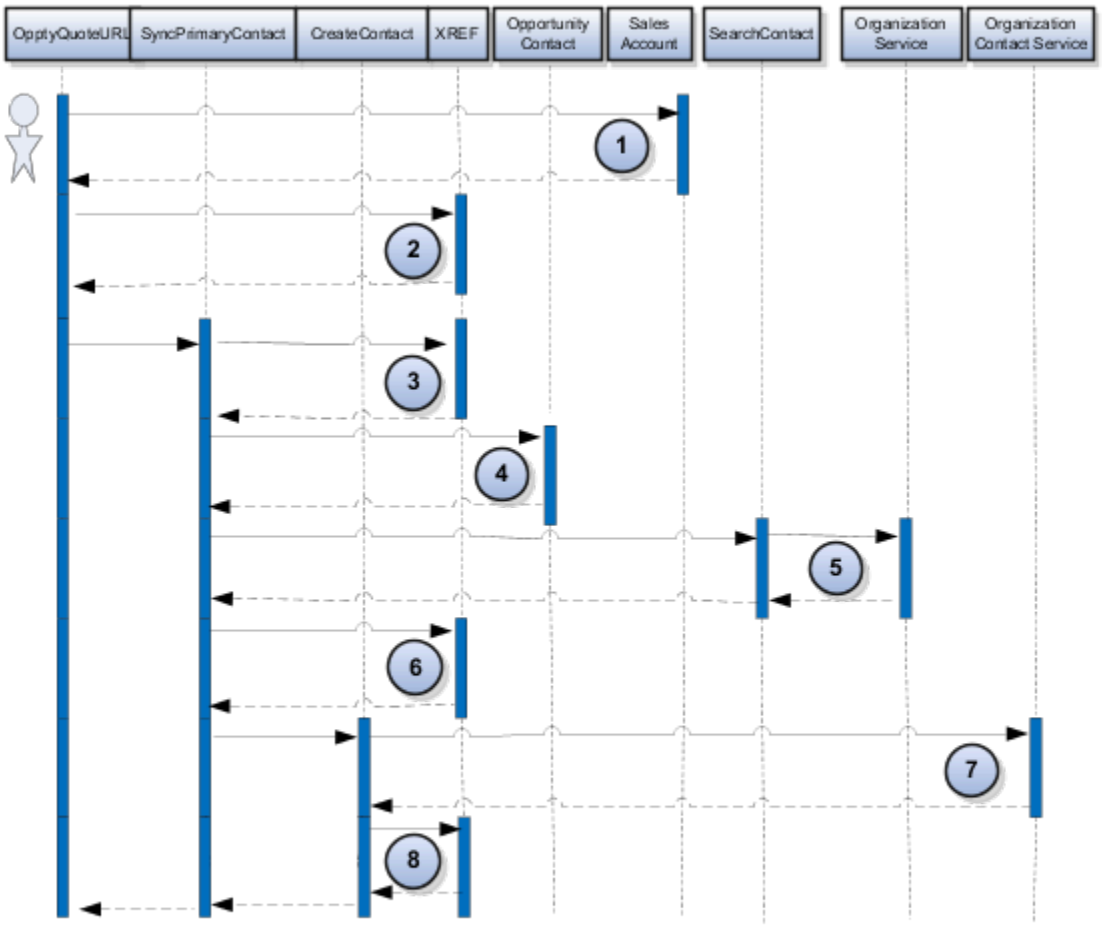
When the user clicks the Quotes tab or the Create Quote button in an opportunity in Oracle CX Sales, the primary contact assigned to the opportunity is compared with organization records in Oracle E-Business Suite. The primary contact is a contact in Oracle CX Sales that has a null cross-reference in Oracle E-Business Suite.

The process retrieves the entire organization record from Oracle E-Business Suite and loops through all of the contact records in the organization record. If it finds a single match, then it creates a cross-reference for the contact in Oracle E-Business Suite: it stores the local identification for the contact in Oracle CX Sales as the remote identification in Oracle E-Business Suite.

If the process does not find a match, or if there are multiple potential matches, then it calls the Create Organization Service web service in Oracle E-Business Suite. It also stores the local identification of the contact in Oracle CX Sales as the remote identification in Oracle E-Business Suite.

Detailed Contact Data Matching Process

The following figure details the contact data matching sequence.



When a salesperson clicks the Quote tab or the Create Quote button in an opportunity in Oracle CX Sales:

1. The process returns the Account object from Oracle CX Sales, and determines the synchronization status.
2. If the customer was synchronized previously, then the process returns the remote customer identification from the XREF custom object. The remote customer identification already exists in Oracle E-Business Suite because of a previous synchronization.
3. The process reads the remote contact identification from the XREF object.
4. The process fetches the primary contact from the Opportunity Contact object in Oracle CX Sales.
5. The process calls the GetOrganization method in the Organization Service web service in Oracle E-Business Suite using the remote customer identification as an input argument.
6. If the returned organization has a matching contact, then the process creates an XREF object with the local and remote contact identifications.
7. If there are no matching contacts or more than one matching contact, then the process calls the createOrgContact method in the Organization Contact Service web service in Oracle E-Business Suite. The createOrgContact method returns the remote contact identification from Oracle E-Business Suite.
8. The process creates an XREF object using the local and remote contact identifications.

Quote Creation in Oracle E-Business Suite

This process lets you synchronize quote data with Oracle E-Business Suite.

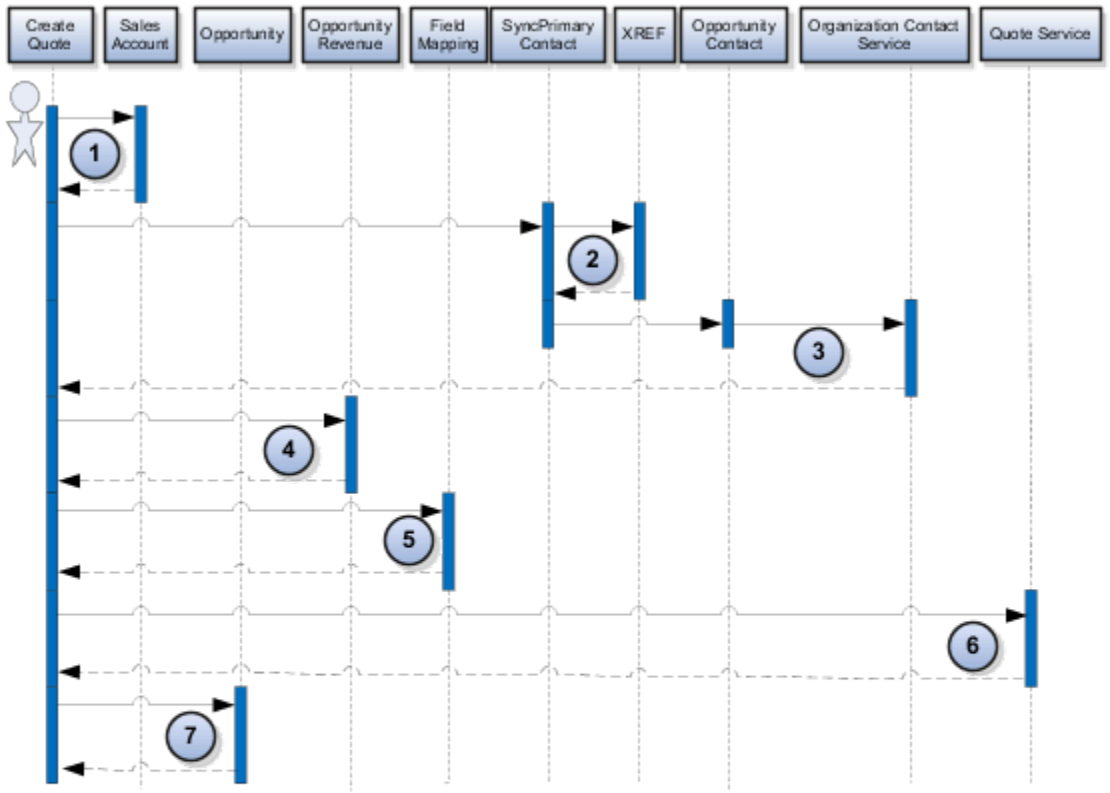
After the customer record is created in Oracle CX Sales and synchronized with Oracle E-Business Suite, the Oracle E-Business Suite quote UI can be accessed. The quote UI is embedded within a tab in the Opportunity Detail screen in Oracle CX Sales.

This embedded quote UI allows the user to take advantage of Oracle E-Business Suite directly in Oracle CX Sales. The quotes, orders, products, pricing, and configuration details are captured in Oracle E-Business Suite. The process also maps revenue line items in opportunity records in Oracle CX Sales to quote line items in Oracle E-Business Suite.

Detailed Quote Creation Process

The following describes the seven-step quote creation process.

The following figure displays the seven-step quote creation sequence.



When a salesperson clicks Create Quote on an Opportunity page in Oracle CX Sales:

1. The process returns the Account object in Oracle CX Sales, and determines the synchronization status.
2. If the customer is synchronized, then the process reads the remote contact identification from the XREF object. If the customer is not synchronized, then an error is returned.
3. If the remote contact identification is not found, then the contact synchronization process is executed.
4. The quote creation process collects opportunity details such as revenue line items from the Opportunity Revenue object.
5. The quote creation process attempts to map all of the revenue lines in the opportunity record to Oracle E-Business Suite products.
6. The quote creation process calls the Create_Quote method in the Quote Service web service in Oracle E-Business Suite, which returns the quote number.

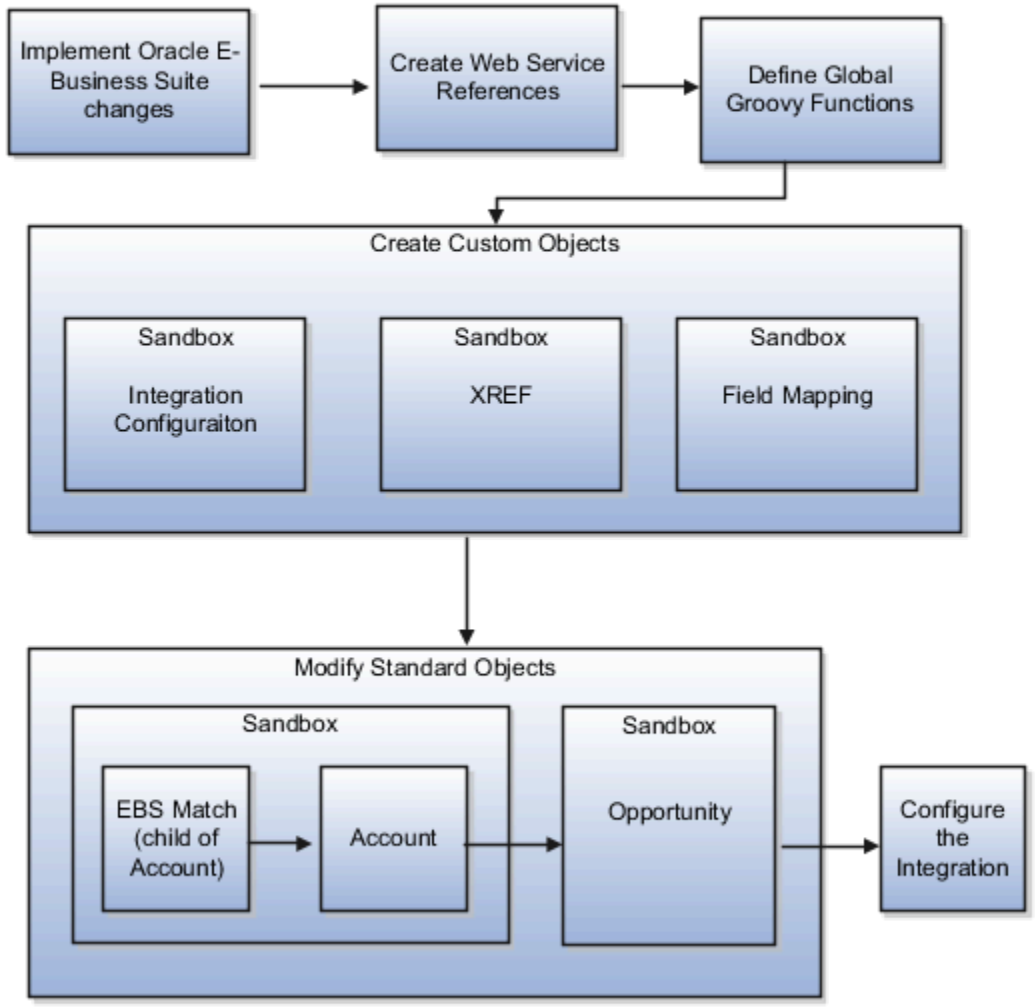
7. The quote creation status, which includes the quote number, is updated in the opportunity. This status does not persist with the opportunity.

Configuration Roadmap

To use this integration, you must configure application artifacts in both Oracle E-Business Suite and Oracle CX Sales.

In the Oracle E-Business Suite instance several patches must be applied, and web services must be configured and exposed. In the Oracle CX Sales instance, custom Groovy logic must be added, custom objects created, and standard objects enhanced. These steps must be executed in order, because many of them rely on the successful completion of previous steps. The following figure shows a high-level overview of the configuration sequence.

The following figure displays a high-level overview of the configuration sequence.



Many of the Oracle CX Sales configuration steps in Chapter 3 have several component procedures, such as creating sandboxes, creating or augmenting objects, defining fields, defining business logic with functions, defining rules or triggers, and defining user interface changes. These component procedures are used in many different configuration steps, but not all procedures are used in all steps. For example, the procedure to create an object or the process to define a trigger is the same from one step to the next, but the names and details of the objects or triggers will change from step to step.

The general process for each component procedure is listed before the specific objects that are to be created or configured. The topics for the specific objects provide details on each of the configuration steps, and which component procedures and details are required in each step.

3 Oracle E-Business Suite Configuration

Software Requirements for Oracle E-Business Suite

The supported Oracle E-Business Suite version for the integration is 12.1.3.

Your Oracle E-Business Suite environment must have the following features for the integration to work:

- Oracle Quoting Application
- Oracle Trading Community Architecture
- Oracle Trading Community Architecture Data Quality Management
- Oracle E-Business Suite Integrated SOA Gateway

  For the list of supported web browsers for Oracle E-Business Suite Release 12, see 389422.1 (Article ID) on My Oracle Support.

Applying Oracle E-Business Suite Patches: Explained

You must apply the patches listed in the following table to your Oracle E-Business Suite environment before performing any configuration steps.

Patch	Description	Status
13391839:R12.ASO.B	POST 12.1.3 ORDER CAPTURE (ASO) CUMULATIVE PATCH	GA
16794688:R12.QOT.B	INTERMEDIA SEARCH RETURNS NO RESULTS WITH ASO_WEB_ATTR_FILTER=YES	GA
16995310:R12.ASO.B	DIRECT ITEM ENTRY IS CLEARING ASO_ QUOTE_LINES_ALL COLUMNS UPON UPDATE	GA
17025892:R12.QOT.B	FUSION CRM OPPORTUNITY INTEGRATION	GA
9328600:R12.OWF.B	12.1.3: REMOVE WSIF AND USE ORASAAJ IN SIF	GA
16872352:R12.OWF.B	1OFF:12.1.3:ADD NONCE AND CREATION TIME TO WSSE SECURITY HEADER FOR USERNAME TOKEN	GA
17591280:R12.QOT.B	FCRM INTEGRATION - PASS ORG CONTACT ID INSTEAD OF RELATIONSHIP PARTY ID	GA

Patch	Description	Status

Create a Generic Error Page

You must create a custom, static error page. Oracle CX Sales will point to this error page if a user experiences an error while trying to access the correct view in the Oracle E-Business Suite Quotes module.

To create a generic error page, do the following:

1. Upload the following HTML file to the Oracle E-Business Suite application server: O_INT_EBS_ErrorPage.html.
2. Source the Oracle E-Business Suite applications environment file, if not already sourced.
3. Copy the uploaded HTML file to the **$OA_HTML** directory.

Assigning Roles to the Oracle E-Business Suite User: Explained

The following roles must be assigned to the Oracle E-Business Suite user who will be used to authenticate the web services required by the integration:

- Trading Community Manager
- Customers Online Superuser
- Quoting User

To assign roles to the Oracle E-Business Suite user do the following:

1. Log in to Oracle E-Business Suite as a user with the system administrator role.
2. Navigate to **User Management,** then **Users**.
3. Search for the user.
4. In the search results, click the **Update** icon in the row for the user.
5. On the **Roles** tab, scroll through the required roles to confirm that they have been assigned to the user.
6. For each required role that has not been assigned, do the following:
    a. Click **Assign Roles**.
    b. In the **Search and Select: Assign Roles** dialog box, search for the role by **Roles and Responsibilities**.
    c. In the search results, select the check box for the role.
    d. Click **Select**.

Expose Web Services

For Oracle CX Sales to be able to call the Oracle E-Business Suite web services that you will be deploying, the web services will need to be exposed securely to the Internet.

The Oracle CX Sales to Oracle E-Business Suite integration requires that published web services be secured using Secure Sockets Layer (SSL). For more information, see the topic on Secure Sockets Layer Configuration in Oracle E-Business Suite System Administrator's Guide - Configuration.

- Use a valid signed server certificate in the setup, not a self-signed certificate. Without a valid certificate Oracle CX Sales will not be able to call Oracle E-Business Suite web services.
- Use the standard HTTPS port (443) when enabling SSL in Oracle E-Business Suite so that Oracle CX Sales can call the web services.
- Do not disable HTTP access if you want users to be able to access Oracle E-Business Suite using standard HTTP from the internal network or over VPN. After configuring SSL, the firewall rules must be configured to allow your Oracle CX Sales instance to make inbound calls to the Oracle E-Business Suite web services.
- Do not allow inbound calls over the standard HTTP port (80).
- It is recommended to allow access only to the web service URLs for the web services specified in this guide.
- If possible, it is also recommended to only allow access from the Oracle CX Sales instance you are using.
- Enable these restrictions after testing successful access to the web services from Oracle CX Sales.

Deploy Web Services

You must deploy the required Oracle E-Business Suite web services which are listed in the following table. Oracle CX Sales calls these web services directly.

The web services must also be granted to the Oracle E-Business Suite user with the roles specified in the in the Assigning Roles to the Oracle E-Business Suite User topic.

> **Note:** When logging into Oracle E-Business Suite, make sure you use the HTTPS URL so that the generated Web Services Description Language files (WSDL) will also use HTTPS in their internal references

The following table lists supported web services, their internal name, type and associated methods.

Web Service	Internal Name	Type	Methods
DQM Search Service	/oracle/apps/ar/hz/service/party/DqmSearchService	Business Service Object	findParties
Organization Service	/oracle/apps/ar/hz/service/party/	Business Service Object	createOrganization getOrganization
Organization Contact Service	OrgContactService	Business Service Object	createOrgContact
Quote Service	ASO_QUOTE_PUB	PL/SQL	Create_Quote

To deploy web services do the following:

1. Log into Oracle E-Business Suite as a user with the system administrator role.
2. Navigate to **Integrated SOA Gateway**, then **Integration Repository**.

3. For each of the web services listed in the preceding table, perform the following steps if the web service has not already been deployed:
    a. Click **Search**.
    b. Enter the internal name and type, then click **Go**.
    c. In the search results, click the link for the web service..

       The web service details are displayed.
    d. Click **Regenerate WSDL**.
    e. In the **Methods** area, select the correct method, and then click **Create Grant**.
    f. On the **Create Grants** page, enter the user's name in the format LastName, FirstName.
    g. In the search results pop-up menu that appears, click the user's name.

       The page refreshes itself.
    h. Click **Apply**.
    i. Repeat the previous steps for other methods as necessary.
    j. In the **Web Service - Web Service Provider** area, select the **User name Token** check box, and then click **Deploy** for all web services.

Related Topics

- Assigning Roles to the Oracle E-Business Suite User: Explained

Create a Quotes Menu

You create a new Quotes menu to display Oracle E-Business Suite quotes in Oracle CX Sales, then a new responsibility to allow the new menu to be displayed. Use the following tasks to create a Quotes menu.

Creating the Quotes Menu

You create a quotes menu for the integration by copying the existing Quoting User menu.

1. Log into Oracle E-Business Suite as system administrator.
2. Navigate to **Function Developer**, **Core Services**, and then **Menus**.
3. Search for the menu code **QOT_QUOTE_ROOT**.

   This is the menu code for the Quoting User menu.
4. Click **Duplicate**.
5. Fill in the fields listed in the following table, and then click **Apply**.

Field	Value
Name	QOT_QUOTE_ROOT_INT
Code	QOT_QUOTE_ROOT_INT

Field	Value
Description	Quote Root Menu for integration

6. Click **Add Another Row**.
7. Add the function displayed in the following table, and then click **Apply**.

Field	Value
Function Name	Quote List
Function Code	QOT_OAUI_QUOTE_LIST

8. Make sure that the function QOT_OAUI_QUOTE_LIST is not on the menu QOT_OAUI_OTHER_MENU by doing the following:
    a. In **Menus**, search for **QOT_OAUI_OTHER_MENU**.
    b. If the Quote List function, with code **QOT_OAUI_QUOTE_LIST**, appears on the list, then remove it.

Creating the Integration Responsibility

Next, you create a responsibility for using the QOT_QUOTE_ROOT_INT menu. The responsibility must be added to all Oracle CX Sales users who will use the Oracle E-Business Suite quoting functionality.

To create the integration responsibility for the menu, do the following:

1. Click Navigator, System Administrator, Security, Responsibility, and then **Define**.

   The form-based Oracle Applications UI appears, with the Responsibilities form active.
2. Click **New**.
3. Populate the fields with the values from the following table, and then click **Save**.

Field	Value
Responsibility Name	Integration Responsibility
Application	Quoting
Responsibility Key	FND_QUOTE_INT
Description	Quoting responsibility for integration
Data Group Name	Standard

Field	Value
Data Group Application	Quoting
Menu	QOT_QUOTE_ROOT_INT

Adding the Integration Responsibility to a User

Finally, you add the new responsibility to an existing user. This user will use the responsibility when launching the Quoting application from Oracle CX Sales.

To add the integration responsibility to an existing user do the following:

1. In the Navigator - System Administrator window, click the **Functions** tab.
2. In the Top Ten List, click **Users**.
3. In the Users form, click **Find**.
4. In the Users dialog box, search for the menu user, for example, LJONES, then click **OK**.
5. In the Users form, click the **Direct Responsibilities** tab, and then scroll to the end of the list.
6. Click **New**.

   A new row appears in the Direct Responsibilities area.
7. Click the ellipsis in the new row.

   The Responsibilities dialog appears.
8. Search for **Integration Responsibility**, and then click **OK**.

   After logging out, and then logging back in as the menu user, you will see Integration Responsibility in the main menu.

4 Oracle CX Sales Configuration

Before You Configure Oracle CX Sales

Before configuring Oracle CX Sales, you must perform the following tasks in your Oracle E-Business Suite environment.

Related Topics

- Oracle E-Business Suite Patching Procedures

Software Requirements for Oracle CX Sales

The integration works with Oracle CX Sales.

For the list of supported web browsers for Oracle CX Sales, see: https://www.oracle.com/system-requirements/index.html.

Related Topics

Create References to Oracle E-Business Suite Web Service

You must create Oracle CX Sales references to the Oracle E-Business Suite web services required for the integration. This is performed in the Application Composer.

When registering the first web service reference in Oracle CX Sales, you must create a new credential key, which Oracle CX Sales will use to authenticate the web service calls to Oracle E-Business Suite. Choose the Oracle E-Business Suite user that you created in the Assigning Roles to the Oracle E-Business Suite User task. All web service calls will be executed by authenticating as this user.

> **Note:** You must also grant the Oracle E-Business Suite user explicit access to the web services as done in the Deploying Web Services task, using the Integration Repository user interface in Oracle E-Business Suite.

Creating Web Service References for the Sales Application

You create web service references for the Sales application using Application Composer. Use the following task to create web service references:

1. Click Navigator, and select **Application Composer**.

2. Click the **Create Web Service Reference** icon on the Web Service page..
3. For connection type, select SOAP or REST, and then click **OK**.
4. Enter the name and WSDL for the DQM Search Service web service reference as shown in the following table, and then click **Read WSDL**.

> **Note:** If a warning appears with the following message: "The preferred security options could not be determined. Contact your web service provider or consult your web service documentation to obtain the correct security details. If errors occur during service invocation, contact your system administrator with the web service and error details" click **OK** to continue.

Field	Value
Name	O_INT_EBS_DQMSearchService
WSDL	https://<EBS host;port>/webservices/AppsWSProvider/oracle/apps/ar/hz/service/party/DqmSearchService?wsdl

5. Click the Add (**+**) button, then enter the information listed in the following table:

Field	Value
Credential Key	Enter the credential key using the following format: EBS_user_name_KEY.
User name	Enter the Oracle E-Business Suite user name.
Password	Enter the Oracle E-Business Suite password.

6. Select the new credential key from the **Credential Key** drop-down list.
7. Fill in or confirm the following fields listed in the following table, and then click **Save and Close**.

Field	Value
Service	DqmSearchService_Service.
Port	DqmSearchService_Port.
Security Scheme	Call with separate user credentials over SSL.
Disable time stamp verification	Select the check box.

8. Repeat the previous steps for each of the following web services.

a. The following table lists the fields and values required to create the OrganizationService

Field	Value
Name	O_INT_EBS_OrganizationService
WSDL	https://EBS host:port name/webservices/AppsWSProvider/ oracle/apps/ar/hz/service/party/OrganizationService?wsdl
Service	OrganizationService_Service
Port	OrganizationService_Port
Security Scheme	Call with separate user credentials over SSL
Credential Key	EBS_user_name_key
Disable time stamp verification	Select the check box

b. The following table lists the fields and values required to create the Organization Contact Service

Field	Value
Name	O_INT_EBS_OrgContactService
WSDL	https://EBS host:port name/webservices/AppsWSProvider/ oracle/apps/ar/hz/service/party/OrgContactService?wsdl
Service	OrgContactService_Service
Port	OrgContactService_Port
Security Scheme	Call with separate user credentials over SSL
Credential Key	EBS_user_name_key
Disable time stamp verification	Select the check box

c. The following table lists the fields and values required to create the Quote Service

Field	Value
Name	O_INT_EBS_QuoteService
WSDL	https://EBS host:port name/webservices/SOAProvider/ plsql/aso_quote_pub/?wsdl
Service	ASO_QUOTE_PUB_Service
Port	ASO_QUOTE_PUB_Port
Security Scheme	Call with separate user credentials over SSL
Credential Key	EBS_user_name_key
Disable time stamp verification	Select the check box

Related Topics

- Assigning Roles to the Oracle E-Business Suite User: Explained
- Expose Web Services
- Deploy Web Services

Create and Activate Sandboxes

To make changes to the application, you must first store the changes in an active sandbox. You can either create a sandbox or select an existing one, and make it active.

Watch video

You must activate the configuration tools you want to use in your sandbox. If you plan to use Page Composer in your sandbox and edit pages at a layer other than Site, you need to create a sandbox just for that layer, and activate only Page Composer in it.

Create and Activate Sandboxes

Follow these steps to create and activate sandboxes for most configuration tools. For flexfields, use the **Manage Descriptive Flexfields** task or the **Manage Extensible Flexfields** task instead.

1. Click **Navigator** > **Configuration** > **Sandboxes**.

2. On the Sandboxes page, click **Create Sandbox**.
3. Enter a name and description for your sandbox.
4. In the **Publishable** field, select **Yes** or **No**. If you set this option as **No**, you can just use your sandbox for testing purposes, but can never publish it.
5. In the All Tools section, select the tools you want to activate for this sandbox. The context layers for all selected tools are set as **Site** by default. So the changes you make using these tools affect all users.
6. If you select Page Composer, you can click the **Edit Sandbox Context** icon and change the context layer from **Site** to another layer, for example **Internal**. You can find the **Edit Sandbox Context** icon in the Support Context column. Make sure you select a context that's supported by the page you want to edit. Otherwise, you won't be able to edit the page.

   > **Note:** If you want to use other tools along with Page Composer in your sandbox, don't change the context layer for Page Composer, even though you can. That's because all tools except Page Composer support only a single context layer, Site. So if you change the context layer for Page Composer from Site to any other layer, all other tools that you might have selected earlier will be deselected.

7. Click **Create** to just create the sandbox, or **Create and Enter** to enter or activate the sandbox after creating it.

Here are a few things to know about activating tools in your sandbox.

- If you try to use a configuration tool in a sandbox without activating the tool in it, you get a message prompting you to activate the tool. You can add more tools to your sandbox later also.
- To create and manage saved searches and make UI adjustments (for example, change a table's column width) just for yourself, you must leave your sandbox before making these changes. But if you want to make these changes for others too, then make the changes with Page Composer open, in which case you also must be in a sandbox.

Activate Existing Sandboxes

Follow these steps to activate a sandbox.

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. From the list of sandboxes, if available, find the one you want to activate, and click the **Enter Sandbox** icon for that sandbox. Your sandbox is activated, and you can see its name on the sandbox bar before the global header. You can use the options available on the sandbox bar to quickly do some activities, such as view sandbox details, publish the sandbox, or leave the sandbox.

Publish Sandboxes

After you're done making changes to the application, publish the sandbox to make your changes available to all users. You must have the Administer Sandbox (FND_ADMINISTER_SANDBOX_PRIV) privilege to publish sandboxes.

> **Note:** Remember, you can't make further changes in the sandbox once you publish it.

Before you start, do these tasks:

- Test or validate your changes in the sandbox in preview mode before actually publishing it. If you made changes using Page Composer, don't forget to close it before testing. To preview your changes, on the sandbox bar before the global header, click **Sandbox Mode**, and select **Preview as if Published (Context: All)**.

> **Note:** You can see the sandbox bar only when you're in an active sandbox.

- Resolve all conflicts flagged in the merge log of your sandbox.

To publish a sandbox:

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. On the Sandboxes page, click the name of the sandbox you want to publish.
3. Click **Publish**.

> **Note:** You might not be able to publish your sandbox because of various reasons. For example, you haven't yet made any changes in your sandbox, or the **Control Publish Sandbox Action in Production Environment** profile option (FND_ALLOW_PUBLISH_SANDBOX) is set to **No**.

4. Click **Continue** to Publish. The sandbox is published to the mainline metadata.
5. Click **Done**.

Related Topics

- Create and Activate Sandboxes

Overview of Functions

There are two types of functions:

- **Global Functions**. Global functions work in all applications in Oracle CX Sales that have extensibility enabled. They are created in Application Composer.
- **Object Functions**. Object functions are attached to a specific object. They are created in the Object Functions tab of the Server Scripts section of the object definition in the Application Composer.

  Each function for the integration is defined in a Groovy script file. At the beginning of each file, a set of comments are listed in the following order:
- Function name
- Function return type
- Function parameters (one per line)

  Following the comments, the body of the function is listed.

Overview of Global Functions

The global functions for the integration are defined in the following Groovy script files:

- O_INT_GetLogMsg.groovy
- O_INT_GetSysParam.groovy
- O_INT_AddMultiValueCriteriaItem.groovy

- O_INT_ApplyFilter.groovy
- O_INT_Debug.groovy
- O_INT_Error.groovy
- O_INT_FindRowByKey.groovy
- O_INT_GetLogMsg.groovy
- O_INT_GetRecordCount.groovy
- O_INT_GetRecords.groovy
- O_INT_EBS_GetSysParam.groovy
- O_INT_Info.groovy
- O_INT_Log.groovy
- O_INT_LogMessage.groovy
- O_INT_Warn.groovy

The content of the Groovy files must be entered into Oracle CX Sales using the Application Composer. Before setting up global functions, create a sandbox and activate it. For more information, see the Create and Activate Unified Sandboxes topic.

Creating Global Functions: Explained

To create a global function, do the following:

1. Click **Navigator**, and select **Application Composer**.
2. In the Overview window, click **Global Functions**.
3. For each Groovy script file do the following:
    a. Select the row, then click the **Action** drop-down list, and select **Add**.

       The Create Global Function window appears.
    b. In the **Function Name** field, enter the function name from the first comment in the Groovy file.
    c. From the **Returns** drop-down list, select the return type specified in the second comment in the Groovy file.
    d. In the Parameters list, click the **Action** drop-down list, then click the **Add** icon.
    e. Enter the name and type for the first parameter listed in the comments in the Groovy file.
    f. Repeat the steps for the remaining parameters.
    g. Copy the Groovy code from the file into the Function Body window.
    h. Click **Validate** to validate the function.

       > **Note:** When validating, certain functions will trigger a warning. This is expected; you can click OK to continue. However, if an error is triggered, then contact Oracle Global Customer Support.

    i. Click **Save and Close**.

Creating Object Functions: Explained

The object functions for the integration are defined in Groovy script files. These files are listed along with the specific objects that must be created or configured. Use the following general procedure to create object functions.

1. Click **Navigator**, and select **Application Composer**.
2. Expand the object for which you want to create a function, and then click **Server Scripts**.
3. In the Server Scripts window, click the **Object Functions** tab.
4. For each Groovy script file do the following:
    a. Click the **Action** drop-down list, and select **Add**.
    The Create Object Function window appears.
    b. In the **Function Name** field, enter the function name from the first comment in the Groovy file.
    c. From the **Returns** drop-down list, select the return type specified in the second comment in the Groovy file.
    d. In the **Parameters** list, click the **Add Parameter** icon.
    e. Enter the name and type for the first parameter listed in the comments in the Groovy file.
    f. Repeat the previous steps for the remaining parameters.
    g. Copy the Groovy code from the file into the **Function Body** window.
    h. Click **Validate** to validate the function.

    > **Note:** When validating, certain functions may trigger a warning. This is expected; you can click **OK** to continue. However, if an error is generated, then contact Oracle Global Customer Support.

    i. Click **Save and Close**.

Create Validation Rules

Validation rules are created in the Validation Rules tab of the Server Scripts section of the object definition in Application Composer.

Validation rules for the integration are defined in Groovy script files. At the beginning of each file, a set of comments is listed in the following order:

- Name
- Description
- Error message

Following the comments, the body of the rule is listed. The Groovy files are displayed as rules to be created for specific objects. Use the following general procedure to create validation rules for objects.

1. Click Navigator, and then select **Application Composer**.
2. Expand the object for which you want to create a validation rule, and then click **Server Scripts**.
   In the Server Scripts window, the Validation Rules tab is shown by default.
3. In the Object Rules area, click the **Add a new validation rule** icon.

4. In the Create Object Validation Rule window, in the **Name** field, enter the rule name and description from the comments in the Groovy file.
5. In the **Error Message** field, enter the error message from the comments in the Groovy file.
6. Copy the Groovy code from the file into the **Rule Definition** window.
7. Click **Save and Close**.

Create Custom Objects

Create a sandbox for each custom object.

After creating each custom object, validate the object and publish the sandbox before moving on to the next custom object. Only create a new sandbox after publishing the previous one. Never have two sandboxes active at the same time.

You create custom objects in the Application Composer.

Use the following procedure to create a custom object:

1. Click **Navigator**, and then select **Application Composer**.
2. In the Custom Objects list, click the **Create a New Object** icon.
3. In the Create Custom Object dialog box, enter the object information, and then click **OK**.

   The custom object appears in the Custom Objects list in the Objects menu.

Adding Fields to a Custom Object

You add fields using the Fields link nested within the custom object in the Application Composer. Use the following procedure to add fields to a custom object.

1. Expand the custom object's node in the Custom Objects list in the **Objects** menu.
2. Click **Fields**.
3. On the Fields page, click the **Create a custom field** icon.
4. In the **Select Field Type** dialog box, select the appropriate field type, and then click **OK**.
5. In the Create <Object Type> Field window, enter the parameters, and then click **Save and Close**.
6. Repeat the previous steps for other fields as necessary.

Creating Pages for a Custom Object

After you create a custom object and add fields to it, you create pages to expose the new object and its fields to users. Every top-level Oracle CX Sales object has an overview page, a creation page, and a details page. These pages make up a work area. You use an object's Pages area in Application Composer to create pages. Use the following procedure to create pages for a custom object.

1. Expand the custom object's node in the **Custom Objects** list in the **Objects** menu.
2. Click **Pages**.
3. On the landing page, click **Edit Summary Table** then do the following:
   a. In the **Configure Summary Table** area, add the fields that you want to display in the Summary Table to the Selected Fields list.
   b. In the **Configure Summary Table: Buttons and Actions** area, make sure Create is in the Selected Buttons list.
   c. Click **Save and Close**.
4. In the **Creation Page Layouts** area, click the **Default Custom Layout** link, and then do the following:

a. Click the **Edit** icon.
b. In the **Configure Detail Form** area, add the fields you want to display in the detailed summary.
c. Click **Save and Close**.
d. In the Creation Layout: Standard layout page, click **Done**.

5. In the **Details Page Layouts** area, select click the **Standard layout** link, then do the following.

a. On the Details Layout: Standard layout, in the Summary area, click the **Edit** (pencil) icon.
b. In the Configure Detail Form area, select the fields you want to display in the default summary.
c. Click **Save and Close**.
d. In the Details Layout: Standard layout page, click **Done**.

Create the Integration Configuration Custom Object

You create the Integration Configuration custom object in the Common application. Use a new sandbox to create the Integration Configuration object.

Enter the parameters listed in the following table for the Integration Configuration custom object, using the Creating Custom Objects procedure.

Parameter	Value
Display Label	Integration Configuration.
Plural Label	Integration Configuration.
Record Name Label	Integration Configuration ID.
Record Name Data Type	Select Automatically Generated Sequence from the drop-down list
Sequence Format	{000000000000000} (15 zeroes inside braces)
Object Name	O_INT_IntegrationConfig
Description	Leave blank.

Creating Fields for the Integration Configuration Custom Object

Create the fields listed in the following table for the Integration Configuration custom object using the Creating Custom Objects procedure.

Parameter	Field: Key	Field: RemoteSystemID	Field: Value
Field Type	Text	Text	Text
Display Label	Key	RemoteSystem	Value
Name	Key	RemoteSystemID	Value
Display Width	20	20	50
Display Type	Simple Text Box	Simple Text Box	Simple Text Box
Constraints	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable
Maximum Length	100	80	500

Creating Pages for the Integration Configuration Custom Object

Create the pages listed in the following tables for the Integration Configuration custom object using the Creating Custom Objects procedure.

The following table lists the required information to configure pages for the Integration Configuration custom object .

Page Layout	Layout Name	Item	Value
Landing Page Layouts	Default Custom Layout	Drill Down Column	Configuration ID
		Selected Fields	• Remote System • Configuration ID • Key • Value
		Buttons and Actions	Select both the Export and Create options.
Creation Page Layouts	Default Custom Layout	Selected Fields	• Integration Configuration ID • Remote System • Key • Value
Details Page Layouts	Default Custom Layout	Selected Fields	• Integration Configuration ID • LastUpdateBy • LastUpdateDate

Page Layout	Layout Name	Item	Value
			• Remote System • Key • Value

Creating a Validation Rule for the Integration Configuration Custom Object

Create the object validation rule from the UniqueKey.groovy file, using the Creating Validation Rules procedure.

Creating Global Functions for the Integration Configuration Custom Object

Create global functions from the following Groovy files, using the Creating Global Functions procedure.

- O_INT_GetIntegConfigParameter.groovy (returns a specific parameter for a category)
- O_INT_GetIntegConfigParameters.groovy (returns all parameters for a category)

> **Note:** These global functions depend on the Integration Configuration custom object being created, so they were not created previously in the Creating Global Functions procedure..

Validating and Publishing the Sandbox

Validate and publish the sandbox for the Integration Configuration custom object, using the Publish Unified Sandboxes topic.

Create the XREF Custom Object

You create the XREF custom object in the Common application. Use a new sandbox to create the XREF custom object.

Creating the Object for the XREF Custom Object

Enter the parameters listed in the following table for the XREF custom object, using the Creating Custom Objects procedure.

Parameter	Value
Display Label	XREF
Plural Label	XREF
Record Name Label	XREF ID
Record Name Data Type	Select Automatically Generated Sequence from the drop-down menu.

Parameter	Value
Sequence Format	{000000000000000} (15 zeroes inside braces)
Object Name	O_INT_XREF
Description	Leave blank.

Creating Fields for the XREF Custom Object

Create the fields listed in the following table for the XREF custom object using the Creating Custom Objects procedure.

Parameter	FusionObjectType Field	FusionRecordID Field	RemoteObjectType Field	RemoteRecordID	RemoteSystemID
Field Type	Text	Text	Text	Text	Text
Display Label	Fusion Object Type	Fusion Record ID	Remote Object Type	Remote Record ID	Remote System
Name	FusionObjectType	FusionRecordID	RemoteObjectType	RemoteRecordID	RemoteSystemID
Display Width	20	20	20	20	20
Display Type	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box
Constraints	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed
Maximum Length	100	50	100	50	80

Creating Global Functions for the XREF Custom Object

Create global functions from the following Groovy files, using the Creating Global Functions procedure.

- O_INT_CreateXREF.groovy
- O_INT_GetAllXREF.groovy
- O_INT_GetXREF.groovy
- O_INT_UpdateXREF.groovy

Note: These global functions depend on the XREF custom object being created, so they were not created previously in the Creating Global Functions procedure.

Validating and Publishing the Sandbox

Validate and publish the sandbox for the XREF custom object, using the Publish Unified Sandboxes topic.

Create a Field Mapping Custom Object

You create the Field Mapping custom object in the Sales application. Use a new sandbox to create the Field mapping custom object.

> **Note:** You must validate and publish the sandbox before generating import and export artifacts for the Field Mapping object. The procedure on generating artifacts appears later in this topic.

Creating the Field Mapping Custom Object

Enter the parameters listed in the following table for the Field Mapping custom object, using the Creating Custom Objects procedure.

Parameter	Value
Display Label	Field Mapping
Plural Label	Field Mappings
Record Name Label	Field Mapping ID
Record Name Data Type	Select Automatically Generated Sequence from the drop down menu.
Sequence Format	{000000000000000} (15 zeroes inside braces)
Object Name	O_INT_FieldMapping
Description	Leave blank.

Creating Fields for the Field Mapping Custom Object

Create the fields listed in the following table for the Field Mapping custom object using the Creating Custom Objects procedure.

Parameter	FusionFieldNameF	FusionFieldValue Field	RemoteFieldID Field	RemoteFieldName Field	RemoteField Value Field	RemoteSystemID Field
Field Type	Text	Text	Text	Text	Text	Text

Parameter	FusionFieldNameF	FusionFieldValue Field	RemoteFieldID Field	RemoteFieldName Field	RemoteField Value Field	RemoteSystemID Field
Display Label	Local Field Name	Local Field Value	Remote Field ID	Remote Field Name	Remote Field Value	Remote System Name
Name	FusionFieldName	LocalFieldValue	RemoteFieldID	RemoteFieldName	RemoteFieldValue	RemoteSystemID
Display Width	20	20	20	20	20	20
Display Type	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box
Constraints	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed	Updatable, Searchable	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed	Required, Updatable, Searchable, Indexed
Maximum Length	100	100	100	100	100	80

Creating Pages for the Field Mapping Custom Object

Create the pages listed in the following tables for the Field Mapping custom object using the Creating Custom Objects procedure.

Creating the Object Validation Rule for the Field Mapping Custom Object

Create the object validation rule from the UniqueKey.groovy file, using the Creating Validation Rules procedure.

Creating the Global Function for the Field Mapping Custom Object

Create a global function from the O_INT_GetMappedField.groovy file using the Creating Global Functions procedure.

Note: This global function depends on the Field Mapping custom object being created, so it was not created previously in the Creating Global Functions procedure.

Validating and Publishing the Sandbox

Validate and publish the sandbox for the Integration Configuration custom object, using the Publish Unified Sandboxes topic.

Generating Artifacts for Import and Export

To support the import and export of a custom object, such as Field Mapping, you must first generate the object artifacts required for both import and export.

Note: Before performing this step, validate the current sandbox and publish it. Generating import and export artifacts in sandboxes is not supported.

To generate artifacts for import and export do the following:

1. Exit the sandbox, click **Navigator**, and select **Application Composer**.
2. In the Generate Import and Export Artifacts area, click **Generate**.
3. Refresh the page if necessary.
4. In the Details window, make sure that the statuses for the Field Mapping custom object are the following:
    - Overall Status: Completed
    - Import Status: Success
    - Export Status: Success

Related Topics

- Create and Activate Sandboxes
- Publish Sandboxes

Create the EBS Match Child Object

You create EBS Match as a child object of the Account standard object.

Note: You perform the Creating the EBS Match Child Object, and Configuring the EBS Match Child Object procedures in the same sandbox, validate the configurations, and then publish the sandbox.

Creating the Child Object

Use the following procedure to create the EBS Match child object. The procedure for creating a child object is similar to the procedure described in Creating a Custom Object, but you also complete the Child Collection Name field to specify the internal name for the set of child object records..

1. Click **Navigator**, and select **Application Composer**.
2. Expand the Standard Objects node, and then select **Account**.
3. On the Account: Overview page, click the **Create Child Object** button.
4. In the Create Child Object dialog box, enter the object information listed in the following table, and then click OK.

Parameter	Value
Display Label	EBS Match
Plural Label	EBS Match
Record Name Label	EBS Match ID

Parameter	Value
Record Name Data Type	Select Automatically Generated Sequence from the drop down menu.
Sequence Format	{000000000000000} (15 zeroes inside braces)
Object Name	O_INT_EBS_Match
Description	Leave blank.
Child Collection Name	O_INT_EBS_MatchCollection

Adding Fields to the Child Object

Create the fields listed in the following table for the EBS Match child object using the following procedure.

1. Click **Navigator**, and select **Application Composer**.
2. Click **Sales Object Tags**.
3. From the Objects menu, expand the Standard Objects node, and then select **Account**.
4. Expand the Child Objects node, then click **EBS Match**in the Account: Overview window, and then expand the EBS Match node from the **Custom Objects** list.
5. Click **Fields**.
6. On the Fields page, click the **Create a custom field** icon.
7. In the **Select Field Type** dialog box, select the appropriate field type, and then click **OK**.
8. In the Create Object_Type Field window, enter the parameters, and then click **OK**.
9. Repeat steps 6 through 8 for other fields as necessary.

The following table lists the additional fields required for creating the EBS Match child object.

Parameter	Address1 Field	City Field	State Field	Province Field	Country Field	MatchScore Field
Field Type	Text	Text	Text	Text	Text	Text
Display Label	Address Line 1	City	State	Province	Country	Match Score
Name	Address1	City	State	Province	Country	MatchScore
Display Width	20	20	20	20	20	20
Display Type	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box
Constraints	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable
Maximum Length	80	80	80	80	80	80

Parameter	Address1 Field	City Field	State Field	Province Field	Country Field	MatchScore Field

The following table lists more fields required to create the EBS Match child object.

Parameter	Selected Field	PartyName Field	PartyID Field	PartyNumber Field	PartyType Field
Field Type	Text	Text	Text	Text	Text
Display Label	Address Line 1	City	State	Province	Country
Name	Address1	City	State	Province	Country
Display Width	20	20	20	20	20
Display Type	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box
Constraints	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable
Maximum Length	80	80	80	80	80

The following table lists more fields required to create the EBS Match child object.

Parameter	PrimaryPhoneCountryCode Field	PrimaryPhoneAreaCode Field	PrimaryPhoneNumber Field	PrimaryEmail Field
Field Type	Text	Text	Text	Text
Display Label	Primary Phone Country Code	Primary Phone Area Code	Primary Phone Number	PrimaryEmail
Name	PrimaryPhoneCountryCode	PrimaryPhoneAreaCode	PrimaryPhoneNumber	Primary Email
Display Width	20	20	20	20
Display Type	Simple Text Box	Simple Text Box	Simple Text Box	Simple Text Box
Constraints	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable	Updatable, Searchable
Maximum Length	80	80	80	80

Adding the Constraint Expression for the Selected Check Box

After creating and saving the Selected check box, you edit it to add an expression to the Updatable constraint. To add an expression to the Updatable constraint for the Selected check box do the following:

1. After adding fields to the EBS Match Child object, click the link for the **Selected** check box on the **Fields** page.
2. In the **Edit Check box Field: Selected** window, click the **Expression Builder** in the Constraints area.

   The Expression Builder appears, with the Functions tab displayed by default.
3. In the expression window, enter the following text: Selected_c == null || Selected_c == 'N'
4. Click **OK**.
5. In the **Edit Check box Field: Selected** window, click **Save and Close**.

Creating the Validation Rule

Create the object validation rule from the UniqueKey.groovy file, using the procedure in Creating Validation Rules.

Creating the Global Function

Create a global function from the O_INT_GetMappedField.groovy file, using the procedure in Creating Global Functions:

> **Note:** This global function depends on the Field Mapping custom object being created, so it was not created previously in Creating Global Functions..

Validating and Publishing the Sandbox

Validate and publish the sandbox for the EBS Match child object, using the Publish Unified Sandboxes topic.

Generating Artifacts for Import and Export

To support the import and export of a custom object, such as Field Mapping, you must first generate the object artifacts required for both import and export.

> **Note:** Before performing this step, validate the current sandbox and publish it. Generating import and export artifacts in sandboxes is not supported.

To generate artifacts for import and export do the following:

1. Exit the sandbox, click **Navigator**, and select **Application Composer**.
2. In the **Common Setup** area, click **Import and Export**.
3. In the **Generate Import and Export Artifacts** area, click **Generate**.
4. Refresh the page if necessary.
5. In the Details window, make sure that the statuses for the **Field Mapping** custom object are the following:
   - Overall Status: Completed
   - Import Status: Success
   - Export Status: Success

Related Topics

- Create and Activate Sandboxes
- Publish Sandboxes

Configuring Standard Objects: Explained

Adding fields to a standard object is very similar to Adding Fields to a Custom Object. To add a field to a standard object do the following:

1. Click **Navigator**, and then select **Application Composer**.
2. Select the object, and then click **Fields**.
3. On the Fields page, click the **Create a custom field** icon.
4. In the **Select Field Type** dialog box, select the appropriate field type, and then click **OK**.
5. In the Create Object_Type Field window, enter the parameters, and then click **OK**.
6. Repeat the previous steps for other fields as necessary.

Adding Actions and Links to an Object

Use the following procedure to add actions and links to an object.

1. Click **Navigator**, and then select **Application Composer**.
2. Select the object, and then click **Actions and Links**.
3. On the Object_Name: Actions and Links page, click the **Create** icon.
4. Enter the display label, name, and description.
5. Select the type, **Action** or **Link**.
6. Select the source:
    - **Script.** Select a method name, or click the **New** icon to enter a new object function.
    - **URL.** Define a URL expression.
7. Click **Save**.
8. Repeat the previous steps to add other actions or links as necessary.

Creating Triggers for an Object

Triggers are created in the Triggers tab of the Server Scripts section of the object definition in the Application Composer. Triggers for the integration are generally defined in Groovy script files. At the beginning of each file, a set of comments is listed in the following order:

- Type (object or field)
- Trigger
- Name
- Error message

Following the comments, the trigger definition is listed. The Groovy files are listed after the triggers to be created for specific objects.

Use the following general procedure to create triggers for objects.

1. Click **Navigator**, and then select **Application Composer**.
2. Expand the object for which you want to create a trigger, and then click **Server Scripts**.
3. In the Server Scripts window, click the **Triggers** tab.
4. Click the **Add a new Trigger** icon for Object Triggers or Field Triggers.

   The Create Object Trigger or Create Field Trigger window appears.
5. Select the trigger from the drop-down list, and then enter the name.

   The trigger and name come from the second and third comments in the Groovy file, respectively.
6. In the **Error Message** area, enter the error message, if any is provided.
7. Copy or enter the Groovy code from the file into the **Trigger Definition** window.
8. Click **Save and Close**.

Configure the Account Standard Object

Create a new sandbox for configuring the Account standard object, using the Create and Activate Unified Sandboxes topic.

Adding Fields

Add the fields shown in the following table to the Account standard object, using the Adding Fields to a Standard Object procedure in the Configuring Standard Objects topic.

The following table lists fields required to create the Account standard object.

Parameter	EBS Sync Status Field
Field Type	Text
Display Label	EBS Sync Status
Name	O_INT_EBS_Sync_Status
Display Width	Leave blank
Display Type	Simple Text Box
Constraints	Updatable (Constraint expression: false)
Maximum Length	25
Default Value	Fixed value = Not Synchronized

Creating Global Functions

Create functions from the following Groovy files, using the Creating Global Functions topic:

- O_INT_EBS_SearchCustomer.groovy
- O_INT_EBS_CreateCustomer.groovy
- O_INT_EBS_GetBaseURL.groovy
- O_INT_EBS_GenerateQuoteURL.groovy

Creating Object Functions

Create functions from the following Groovy files, using the Creating Object Functions topic:

- O_INT_EBS_CheckMatch.groovy
- O_INT_EBS_CleanupMatches.groovy
- O_INT_EBS_CustomerQuoteURL.groovy
- O_INT_EBS_CustomerSync.groovy
- O_INT_EBS_GetSearchCriteria.groovy
- O_INT_EBS_PopulateMatches.groovy

Creating Triggers

Create the following triggers using the Creating Triggers for an Object procedure from the Configuring Standard Objects topic.

- Trigger Type: "Before Update in Database".
  Groovy File: O_INT_EBS_CustomerSyncTrigger.groovy file.
- Trigger Type: "Before Insert in Database".
  Groovy File: O_INT_EBS_CustomerCreateTrigger.groovy file.

Creating the Action

Add the Retry Sync action using information from the following table to the Account object, using the Adding Actions and Links to an Object procedure from the Configuring Standard Objects procedure. .

Parameter	Value
Display Label	Retry Sync
Name	Retry_Sync
Type	Action
Description	Leave blank.

Parameter	Value
Source	Source
Method Name	O_INT_EBS_CustomerSync

Configuring Pages

You modify pages for the Account object in the Application Composer. To configure pages for the account standard object, do the following:

1. Click **Navigator**, and then select **Application Composer**.
2. For the Account object, click **Pages**.
3. In the **Details Page Layouts** area, select **Standard Layout**, and then click the **Duplicate Layout** icon.
4. In the **Duplicate Layout** dialog box, enter the following name for the duplicate layout: **Integration Layout**, and make sure that **Standard Layout** is listed as the **Source Layout**.
5. Click **Save and Edit**.
6. In the **Details Page Layouts** area, click the **Standard Layout** link.
    a. In the **Subtabs Region** area, click the **Profile** icon.
    b. In the **Summary** area, click the Edit icon (pencil) to view the **Configure Detail** form.
    c. In the **Configure Detail Form**, move **EBS Sync Status** to the **Selected Fields** list.
    d. Click **Save and Close**.
7. In the **Details Layout: Integration Layout** page, click the **Edit** icon (pencil).
8. In the **Available Actions** area, move **Retry Sync** from the Available Actions list to the **Selected Actions** list.
9. Click **Save and Close**.
10. In the **Subtabs Region** area, click the **Add** icon to add a new subtab.
11. In the **Create Subtab** page, select **Web Content** and click **Next**.
12. In the **Create Subtab: Web Content** page, enter the information in the following table, and when finished, click **Save and Close**.

Parameter	Value
Display Label	Quotes
Source	URL
URL Definition	O_INT_EBS_CustomerQuoteURL()
Display Icon	An icon is selected by default. Click Change Icon if needed.

Adding the EBS Match Tab

You add the EBS Match tab for the Account object in the Application Composer. To add the EBS Match tab, do the following:

1. Click **Navigator**, and then select **Application Composer**.
2. Click **Pages**.
3. In the **Details Page Layout** section, click the **Integration Layout** that you created in the previous task.
4. In the **Subtabs Region** area, click the **Add** icon to add a new subtab.
5. In the **Create Subtab** page, select **Child or related object** and click **Next**.
6. On the **Create Subtab: Child or Related Object** page,click the **Data Object** drop-down list and select **EBS Match**, then enter the information in the following table.

Parameter	Value
Display Label	EBS Match
Display Icon	An icon is selected by default. Click Change Icon if needed.

7. In the **Configure Summary Table** area do the following move Selected, Party Name, Match Score, Party Id, Party Type, Address Line 1, EBS Match ID to the **Selected Fields** list.
8. In the **Configure Summary Table: Buttons and Actions** area, make sure the **Show Edit** check box is selected, and deselect the Show Create and Show Delete check boxes.
9. Move **Create New EBS Customer** to the **Selected Buttons** list.
10. Validate and publish the sandbox for the Account standard object using the Publish Unified Sandboxes topic.

Related Topics

- Create and Activate Sandboxes
- Publish Sandboxes

Configure the EBS Match Child Object

Now you continue configuring the EBS Match child object in Application Composer.

Creating the Object Function

Create an object function from the ForceCreateNewCustomer.groovy file, using the Creating Object Functions topic.

Creating the Validation Rule for the Child Object

Create a validation rule from the O_INT_EBS_MatchSelectRule.groovy file, using the Creating Validation Rules topic.

Creating Triggers for the Child Object

Create triggers from the O_INT_EBS_CleanupMatchesTrigger.groovy file, using the Creating Triggers for an Object procedure from the Configuring Standard Objects topic.

Adding an Action to the Child Object

Add the Create New EBS Customer action shown in the following table to the EBS Match child object, using the Adding Actions and Links to an Object procedure from the Configuring Standard Objects topic.

Parameter	Value
Display Label	Create New EBS Customer
Name	Create_New_EBS_Customer
Type	Action
Description	Leave blank.
Source	Script
Method Name	ForceCreateNewCustomer

Validating and Publishing the Sandbox

Validate and publish the sandbox for the EBS Match object, using the Publish Unified Sandboxes topic..

Related Topics

- Create and Activate Sandboxes
- Publish Sandboxes

Configure the Opportunity Standard Object

Create a new sandbox for configuring the Opportunity standard object, using the procedure in the Create and Activate Unified Sandboxes topic.

Adding Fields

Add the fields shown in the following table to the Opportunity standard object, using the procedure Adding Fields to a Standard Object from the Configuring Standard Objects topic.

Parameter	Sales Document Status Field
Field Type	Text
Display Label	Sales Document Status
Display Type	Simple Text Box
Name	O_INT_SalesDocumentStatus
Constraints	Updateable (Constraint Expression: false)
Maximum length	150
Default Value	None

Creating Global Functions

Create functions from the following Groovy files, using the procedure in the Creating Global Functions topic.

- O_INT_EBS_CreateContact.groovy
- O_INT_EBS_IsMatchedContact.groovy
- O_INT_EBS_SearchContact.groovy

Creating Object Functions

Create functions from the following Groovy files, using the procedure in the Creating Object Functions topic.

- O_INT_EBS_CreateQuote.groovy
- O_INT_EBS_OpptyQuoteURL.groovy
- O_INT_EBS_SearchContactAndCreateXref.groovy
- O_INT_EBS_SyncPrimaryContact.groovy

Creating the Action

Add the Create Quote action shown in the following table to the Opportunity object, using the procedure Adding Actions and Links to an Object from the Configuring Standard Objects topic.

Parameter	Value
Display Label	Create Quote
Name	Create_Quote
Type	Action

Parameter	Value
Description	Leave blank.
Source	Script
Method Name	O_INT_EBS_CreateQuote

Creating Triggers

Create a trigger from the O_INT_CleanUpStatusTrigger.groovy file using the procedure Creating Triggers from the Configuring Standard Objects topic .

Configuring Pages

You modify pages for the Opportunity object in the Application Composer. To configure pages for the Opportunity standard object do the following:

1. Click **Navigator**, and select **Application Composer**.
2. Expand the **Opportunity** node from the Standard Objects list.
3. Click **Pages**.
4. In the Details Page Layouts area, select **Standard Layout**, and then click the **Duplicate Layout** icon.
5. In the **Duplicate Layout** dialog box, enter the following name for the duplicate layout: Integration Layout, and make sure that **Standard Layout** is listed as the **Source Layout**.
6. Click **Save and Edit**.
7. In the **Details Layout: Integration Layout** page, click the **Edit** icon (pencil).
8. On the **Details Layout: Integration Layout: Buttons and Actions** page, move **Create Quote** from the Available Buttons list to the Selected Buttons list, and then click **Save and Close**.
9. In the **Details Layout: Integration Layout** page, in the **Subtabs Region** area, click the **Edit** (pencil) icon.
10. Move **Sales Document Status** from the Available Fields list to the Selected Fields list, and then click **Save and Close**.
11. In the **Configure Detail Form: Buttons and Actions** area, move **Create_Quote** to the Selected Buttons area, and then click **Save and Close**.
12. In the **Subtabs Region** area, click the **Add** icon to add a new subtab.
13. In the **Create Subtab** page, select **Web Content** and click **Next**.
14. In the **Create Subtab: Web Content** page, enter the information in the following table, and when finished, click **Save and Close**.

Parameter	Value
Display Label	Quotes
Source	URL
URL Definition	O_INT_EBS_OpptyQuoteURL()

Parameter	Value
Display icon.	An icon is selected by default. Click **Change Icon** if needed.

Validating and Publishing the Sandbox

Validate and publish the sandbox for the Opportunity standard object, using the procedures in the Publish Unified Sandboxes topic.

Related Topics

- Create and Activate Sandboxes
- Publish Sandboxes

Connect Oracle CX Sales to Oracle E-Business Suite

You use the Integration Configuration page to allow Oracle CX Sales to exchange data with Oracle E-Business Suite.

Note: The provided reference integration content and the associated Groovy script artifacts use EBS_12.1.3 as the name of the Oracle E-Business Suite instance. You can change this value to your own preferred value (such as EBS_12.2, or myEBSsystem, or any other value) by changing the value of the RemoteSystemID entry in the O_INT_EBS_GetSysParam.groovy file to your preferred value, and by substituting your preferred value with the Remote System values listed in the following table.

To connect Oracle CX Sales to Oracle E-Business Suite do the following:

1. Navigate to **Tools**, and then **Integration Configuration**.
2. On the Integration Configuration page, click the **Create** icon.
3. On the Create Integration page, enter the information for the first key-value pair in the following table, and then click **Save and Close**.
4. Repeat the previous steps for the remaining key-value pairs in the following tables.

Remote System	Key	Value
EBS_12.1.3	host	Host name of your Oracle E-Business Suite instance, for example: myEBS_server.example.com
EBS_12.1.3	port	HTTPS port of your Oracle E-Business Suite instance, for example: 443
EBS_12.1.3	protocol	https
EBS_12.1.3	TCA_NLS_LANGUAGE	American

Remote System	Key	Value
EBS_12.1.3	QOT_NLS_LANGUAGE	American
EBS_12.1.3	ORG_ID	Organization ID for your Oracle E-Business Suite instance, for example: 204
EBS_12.1.3	CUST_MATCH_RULE_ID	62
EBS_12.1.3	PRICE_LIST_ID	Price list ID for your Oracle E-Business Suite instance, which can be found by running the following SQL query:`select list_header_id from qp_list_headers_tl where name like '<Price List Name>%' and language = 'US';` When a quote is created from an Oracle CX Sales opportunity, an attempt is made to map opportunity revenue line items to quote line items in Oracle E-Business Suite. This saves the user from having to re-create line items manually in the quote. The price list ID is used as the default in the mapping and quote creation process, and drives the pricing that appears in the generated quote in Oracle E-Business Suite. After the quote has been generated with the default values, the user can make any necessary changes to the quote, such as the pricing.

5 Postconfiguration Tasks

Map Products

For the integration to populate new quotes with quote line items that come from the revenue lines on associated opportunities, mappings must be created to associate products from the front office Sales application to the back-office ERP system.

Extracting the List of Oracle E-Business Suite Products: Explained

First you use a SQL query to extract the list of products from Oracle E-Business Suite. To extract the list of Oracle E-Business Suite products, do the following:

1. Connect to the Oracle E-Business Suite database.
2. Run the following SQL query:

```
SELECT msiv.inventory_item_id
 ,msiv.organization_id
 ,msiv.concatenated_segments item
 ,msiv.description
 ,msiv.primary_uom_code
 ,msiv.serviceable_product_flag
 ,msiv.service_item_flag
 ,msiv.bom_item_type
 ,msiv.item_type
 ,msiv.service_duration
 ,msiv.service_duration_period_code
 ,msiv.shippable_item_flag
 ,msiv.returnable_flag
 ,msiv.web_status
 ,msiv.orderable_on_web_flag
 ,msiv.customer_order_enabled_flag
FROM mtl_system_items_b_kfv msiv, ibe_ct_imedia_search icis,
 mtl_units_of_measure_vl muv
WHERE msiv.organization_id = icis.organization_id
AND msiv.inventory_item_id = icis.inventory_item_id
AND icis.language = userenv('LANG')
AND msiv.primary_uom_code <> 'ENR'
AND (msiv.bom_item_type = 1 OR msiv.bom_item_type = 4)
AND msiv.primary_uom_code = muv.uom_code
AND (
 ((select fnd_profile.value('ASO_WEB_ATTR_FILTER') from dual) = 'Y'
 AND msiv.web_status = 'PUBLISHED' AND orderable_on_web_flag = 'Y')
 OR
 (select fnd_profile.value('ASO_WEB_ATTR_FILTER') from dual) is
 null OR (select fnd_profile.value('ASO_WEB_ATTR_FILTER') from dual) != 'Y')
 )
AND icis.category_set_id = nvl((select fnd_profile.value('ASO_CATEGORY_SET') from dual),
(SELECT category_set_id FROM mtl_default_category_sets WHERE functional_area_id =7))
```

```
AND msiv.organization_id = <organization_id>;
```

3. You can use the following optional AND condition to filter results based on specific item codes: `AND msiv.concatenated_segments in ('<Item name>')`
4. Save the results to a CSV file, then open in spreadsheet software.

   The item code is listed in the Item column. The item description is listed in the Description column.

Extract a List of Oracle CX Sales Products

Next you extract the list of products from Oracle CX Sales. To extract the list of products, do the following:

1. Log into Oracle CX Sales as a user with the Product Manager role.
2. Click **Navigator**, and then click **Product Information Management** link.
3. Click the **Tasks** icon, and then in the **Item Management** area, click the **Browse Items** link.
4. In the Advanced Search area of the Browse Items page, click the **Add Fields** button.
5. In the Select and Add: Attributes page, click the **Query By Example** icon, and then specify a **Creation Date** value for the attributes.
6. From search result, select the row where **Object Item value** has the **Creation Date Attribute** value you entered, and click **Add** and then click **OK**.

   Search criteria for the Creation Date field is added.
7. From the **Creation Date** drop down menu, select **After**.
8. Select a date that is at least five years ago (for example, 1/1/09).
9. Click the **Search** button. The list of items appears. If required, you can filter by Item and Organization and Item.
10. Click the **Export to Excel** icon.
11. Save the file.

> **Note:** Some spreadsheet tools remove the lead zero in a numeric field. This can affect the item numbers if the numbers are only numeric. Use a basic text editor or verify your item numbers to ensure that they are correctly reflected.

For more information on Oracle CX Sales products and creating sales catalogs, see Related Topics.

Related Topics

Map Products Manually

You can create product mappings by comparing the SQL results from Oracle E-Business Suite to the CSV file from Oracle CX Sales. Here is one example for mapping desktop computers. To map products manually, do the following:

1. In the Oracle E-Business Suite SQL results, filter on descriptions containing Desktop.
2. Note the values in the ITEM and INVENTORY_ITEM_ID columns.
3. In the Oracle CX Sales Excel file, filter on descriptions containing Desktop.
4. Note the values in the Item column.
5. Map the corresponding products to each other.

The following table shows an example:

System	Column	Value
Oracle CX Sales	Description	Sentinel 9000 Desktop
Oracle CX Sales	Item	AS54600
Oracle E-Business Suite	DESCRIPTION	Sentinel Deluxe Desktop
Oracle E-Business Suite	ITEM	AS18947
Oracle E-Business Suite	INVENTORY_ITEM_ID	155

Enter Mappings into Oracle CX Sales

You use the Field Mapping object in Oracle CX Sales to enter product mappings. The example in the following table uses the values from the example in the Mapping Products to Each Other Manually topic.

Note: You can also import the product mappings using Oracle CX Sales File Import function.

To enter product mappings into Oracle CX Sales do the following:

1. Log into Oracle CX Sales as a user with the Administrator role.
2. Navigate to **Tools**, and then **Field Mapping**.
3. On the Field Mappings page, click the **Create** icon.
4. Enter the values shown in the following table, then click **Save and Close**.
5. Repeat steps 3 and 4 for other products.
   The following tables lists sample mapping values.

Parameter	Value
Remote System Name	Remote system you entered in Connecting Oracle CX Sales to Oracle E-Business Suite
Local Field Name	Product
Local Field Value	Reference Number column value in Oracle CX Sales, for example, AS54600
Remote Field Name	Product
Remote Field Value	ITEM column value in Oracle E-Business Suite, for example, AS18947

Parameter	Value
Remote Field ID	INVENTORY_ITEM_ID column value in Oracle E-Business Suite, for example, 155

Related Topics

- Map Products Manually

Map Product Groups

Use the following tasks to create product group mappings from Oracle E-Business Suite to Oracle CX Sales.

Extracting the List of Oracle E-Business Suite Products

First you use the following SQL query to extract the list of products from Oracle E-Business Suite.

1. Connect to the Oracle E-Business Suite database.
2. Run the following SQL query:

```
SELECT msiv.inventory_item_id
 ,msiv.organization_id
 ,msiv.concatenated_segments item
 ,msiv.description
 ,msiv.primary_uom_code
 ,msiv.serviceable_product_flag
 ,msiv.service_item_flag
 ,msiv.bom_item_type
 ,msiv.item_type
 ,msiv.service_duratio
 ,msiv.service_duration_period_code
 ,msiv.shippable_item_flag
 ,msiv.returnable_flag
 ,msiv.web_status
 ,msiv.orderable_on_web_flag
 ,msiv.customer_order_enabled_flag
FROM mtl_system_items_b_kfv msiv,ibe_ct_imedia_search icis,
 mtl_units_of_measure_vl muv
WHERE msiv.organization_id = icis.organization_id
AND msiv.inventory_item_id = icis.inventory_item_id
AND icis.language = userenv('LANG')
AND msiv.primary_uom_code <> 'ENR'
AND (msiv.bom_item_type = 1 OR msiv.bom_item_type = 4)
AND msiv.primary_uom_code = muv.uom_code
AND (
 ((select fnd_profile.value('ASO_WEB_ATTR_FILTER') from dual) = 'Y'
 OR
 ((select fnd_profile.value('ASO_WEB_ATTR_FILTER') from dual) is
 null OR (select fnd_profile.value('ASO_WEB_ATTR_FILTER') from dual) != 'Y')
 )
AND (
 ((select fnd_profile.value('ASO_REQUIRE_SERVICE_REFERENCE') from dual) = 'Y' AND service_item_flag =
 'N')
 OR
 ((select fnd_profile.value('ASO_REQUIRE_SERVICE_REFERENCE') from
 dual) is null OR (select fnd_profile.value('ASO_REQUIRE_SERVICE_REFERENCE') from dual) != 'Y')
```

```
 )
AND icis.category_set_id = nvl((select
fnd_profile.value('ASO_CATEGORY_SET') from dual), (SELECT
category_set_id FROM mtl_default_category_sets WHERE functional_area_id =7))
AND msiv.organization_id = <organization_id>;
```

3. You can use the following optional AND condition to filter results based on specific item codes:`AND msiv.concatenated_segments in <'Item_name>')`
4. Save the results to a CSV file, then open in spreadsheet software.

   The item code is listed in the Item column. The item description is listed in the Description column.

Extracting the List of Oracle CX Sales Product Groups

Use this task to extract a list of product groups from Oracle CX Sales.

1. In Oracle CX Sales navigate to **Setup and Maintenance**.

   Oracle Functional Setup Manager appears.
2. From the Setup drop down menu, select **Sales**.
3. In the Search Tasks box, search for **Manage Product Groups**.
4. Click the **Manage Product Groups** link in the search results.
5. In the Sales Catalog and Products task list, click the **Manage Product Groups** link.
6. In the Manage Product Groups area, click the **View** drop down list, and then click **Columns**, and from that Columns menu click the following columns to enable them:
   - Name
   - Reference Number
   - Active
   - Effective from Data
   - Effective to Date
7. In the Manage Product Groups area click the **Export to Excel** icon to export the list.

Mapping Products to Each Other Manually

You can create product mappings by comparing the SQL results from Oracle E-Business Suite to the CSV file from Oracle CX Sales. The following is one example for mapping desktop computers. To map products manually, do the following:

1. Using the Oracle E-Business Suite SQL results generated in Extracting the List of Oracle E-Business Suite Products, filter on descriptions containing **Desktop**.
2. Note the values in the ITEM and INVENTORY_ITEM_ID columns.
3. In the Oracle CX Sales Excel file, filter on descriptions containing Desktop.
4. Note the values in the Reference Number column.
5. Map the corresponding products to each other.

System	Column	Value
Oracle CX Sales	Product Group Name	WAVE Desktops
	Product Group Reference Number	3-57FC-880

System	Column	Value
Oracle E-Business Suite	DESCRIPTION	Sentinel Deluxe Desktop
	Reference Number	AS18947
	INVENTORY_ITEM_ID	155

Entering Mappings into Oracle CX Sales

You use the Field Mapping object in Oracle CX Sales to enter product mappings. The example listed in the following table uses the values from the example in Mapping Products to Each Other Manually.

Note: You can also import the product mappings with the Oracle CX Sales File Import function. To enter product mappings into Oracle CX Sales, do the following:

1. Log into Oracle CX Sales as a user with the Administrator role.
2. Navigate to **Tools**, and then **Field Mapping**.
3. On the Field Mappings page, click the **Create** icon.
4. Enter the values shown in the following table, then click **Save and Close**.
5. Repeat steps 3 and 4 for other products.
6. Click **Save and Close** when finished.

Parameter	Value
Remote System Name	Remote system you entered in Connecting Oracle CX Sales to Oracle E-Business Suite
Local Field Name	Product Group
Local Field Value	Reference Number column value in Oracle CX Sales, for example, AS54600
Remote Field Name	Product
Remote Field Value	ITEM column value in Oracle E-Business Suite, for example, AS18947
Remote Field ID	INVENTORY_ITEM_ID column value in Oracle E-Business Suite, for example, 155

Guidelines for Hiding Oracle E-Business Suite UI Elements for Streamlined Navigation

This integration relies on a technique known as a UI mashup.

The Oracle E-Business Suite quote user interface (UI) is shown within an Oracle CX Sales UI by embedding it inside an HTML inline frame (iFrame). This means that the user will see both the Oracle CX Sales UI and the Oracle E-Business Suite UI at the same time in the browser.

Navigation Issues with UI Mashups

When combining two applications into one UI, however, certain navigational issues can result. While UI mashups are generally designed to show related content from the two applications (for example, a quote that is related to an opportunity or a customer), it is possible to use the navigational controls in the embedded application to go to a part of the embedded application that has no relationship to the parent application.

In the case of embedding an Oracle E-Business Suite UI inside of an Oracle CX Sales UI, it is theoretically possible to use the Oracle E-Business Suite navigator to go to a completely different module within the Oracle E-Business Suite application that has nothing to do with the related customer or opportunity from Oracle CX Sales, where the embedded UI was being displayed. For example, users with privileges in Oracle E-Business Suite that allow them to create quotes might also be able to navigate to Oracle E-Business Suite Human Resources or Financials pages that have no relationship to the customer or opportunity in Oracle CX Sales.

Limiting Unwanted Navigation

To minimize the possibility that end users can navigate and change the context of the embedded Oracle E-Business Suite UI to something that is not related to the parent Oracle CX Sales UI, the integration relies on a specialized patch in Oracle E-Business Suite that hides the header and footer sections in the quote UI pages, which the user would normally see if logged directly in to Oracle E-Business Suite. This partially prevents users from navigating to other modules within Oracle E-Business Suite. However, this does not completely eliminate the possibility of navigating to something out of context. Within the Oracle E-Business Suite quote UI there are also a variety of hyperlinks and buttons that allow the user to navigate to content and modules that are outside of quoting, and therefore outside the context of the Oracle CX Sales-Oracle E-Business Suite quote UI mashup.

Hiding or Disabling Links and Controls

To help minimize this issue, use the following table which shows links and controls that can be hidden or disabled. Note that disabling these UI elements is not required for the integration to work. Instead, you must determine whether limiting navigational options is important for your user communities. In some cases, user training might be sufficient to avoid any navigational confusion. In other cases, limiting what users can actually navigate to or drill down on will be desirable.

If limiting navigational options is required, the following table shows a number of controls that might be helpful to hide or disable. Note that additional controls not listed might need to be hidden or disabled, depending on what customizations have been applied to the Oracle E-Business Suite environment. This list is a starting point, and is not comprehensive.

The following tables list the Oracle E-Business Suite UI Elements to hide on the Oracle CX Sales Quote Update page.

The following table lists the Summary Area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Summary	Not applicable	Select button beside Contact Name	Rendered: false

The following table lists the Product tab area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Product tab	Not applicable	Submit button: Add Service to Previous Purchase	Rendered: false

The following table lists the Pricing tab area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Pricing tab	Not applicable	Add Adjustment in Product tab	Rendered: false
Not applicable	Not applicable	Column: (QotHdrPrcAdjRN.QotHdrPrcAdjRm(Add Promotion in Product tab	Rendered: false
Not applicable	Not applicable	Add Promotion in Product tab	Rendered: false
Not applicable	Not applicable	Column: (QotHDrPrcPromCdRN.QotHdrProm(	Rendered: false
Not applicable	Not applicable	Add Charge in Product tab	Rendered: false
Not applicable	Not applicable	Column: (QotHdrPrcChrgRN.QotHdrChrgRm(	Rendered: false
Not applicable	Not applicable	Reset Pricing button	Rendered: false
Not applicable	Not applicable	Message Text: Pricing Date	Read only: true
Not applicable	Not applicable	Message list of values: Product Fiscal Classification	Read only: true
Not applicable	Not applicable	Message list of values: Transaction Business Category	Read only: true
Not applicable	Not applicable	Message: Tax Handling	Read only: true

Application Area	Application Subarea	UI Element	Property Change
Not applicable	Not applicable	Message: Exemption Reason	Read only: true
Not applicable	Not applicable	Message Text: Exemption Number	Read only: true

The following table lists the Shipping and Billing tab area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Shipping and Billing tab	Address	Go button after Message: Copy the Quote's Customer, Contact and Address To	Rendered: false
Not applicable	Address	Message: Copy the Quote's Customer, Contact and Address To	Rendered: false
Not applicable	Address, Shipping	• Select button beside Costumer Name • Select button beside Contact name • Select button beside Address	Rendered: false
Not applicable	Address, Billing	Select button beside Customer Name	Rendered: false
Not applicable	Not applicable	Message list of values: Account Number	Read only: true
Not applicable	Not applicable	Select button beside Contact Name	Read only: true
Not applicable	Not applicable	Address select button	Rendered: false
Not applicable	Address, End Customer	Customer Name select button	Rendered: false
Not applicable	Not applicable	Message list of values: Account Number	Read only: true
Not applicable	Not applicable	Contact Name select button	Rendered: false
Not applicable	Not applicable	Address select button	Rendered: false
Not applicable	Shipping	• Message Text: Requested Date	Read only: true

Application Area	Application Subarea	UI Element	Property Change
		• Message: Request Date Type • Message: Shipping Method • Message: Shipment Priority • Message: Demand Class • Message: Freight Terms • Message Text: Shipping Instructions • Message: Free On Board (FOB) • Message Text: Packing Instructions	

The following table lists the Product tab area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Sales Team tab	Team Members	Add Person button	Rendered: false
		Reassign Sales Team button	Rendered: false
		Flow Layout: (QotHdrSlsTmTabRN.QotHdrSlsTmA	Rendered: false
		Message: Role	Read only: true
		Message Check Box: Do Not Reassign	Read only: true
		Remove image in the table in the Remove column	Rendered: false
	Sales Credit, Revenue Credits	Add Person button	Rendered: false
		Remove image in the table in the Remove column	Rendered: false
		Message: Sales Group	Read only: true
		Message: Credit Type	Read only: true
		Message Text: Credit Percent	Read only: true
		Column: (QotHdrSlsCrdtRN.QotRvnRemCol)	Rendered: false

Application Area	Application Subarea	UI Element	Property Change
	Sales Credit, Non-Revenue Credits	Add Person button	Rendered: false
		Remove image in the table in the Remove column	Rendered: false
		Message: Sales Group	Read only: true
		Message: Credit Type	Read only: true
		Message Text: Credit Percent	Read only: true
		Column: (QotHdrSlsCrdtRN.QotNonRvnRemC	Rendered: false

The following table lists the Proposals area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Proposals	Not applicable	Submit button: Create Proposal	Rendered: false

Hiding or Disabling Links and Controls Continued

The following table lists the Approvals area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Approvals	Not applicable	Initiate Approvals button	Rendered: false
Not applicable	Not applicable	Cancel Approval Process button	Rendered: false
Not applicable	Not applicable	Table Actions	Rendered: false

The following table lists the Contract Terms area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Contract Terms	Not applicable	Preview Terms button	Rendered: false
Not applicable	Not applicable	Update Terms button	Rendered: false

Application Area	Application Subarea	UI Element	Property Change
Not applicable	Not applicable	Message Text: Signed By	Read only: true
Not applicable	Not applicable	Message Text: Signed Date	Read only: true

The following table lists the Supplement area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Supplement	Not applicable	Update button	Rendered: false
Not applicable	Not applicable	Row Layout: (QotHdrSuppTabRN.QotSupUpdateF	Rendered: false
Not applicable	Not applicable	Remove button	Rendered: false
Not applicable	Not applicable	Message: Section	Read only: true
Not applicable	Not applicable	Message: Template	Read only: true

The following table lists the Notes and Tasks area and the UI element within that area that must be changed.

Application Area	Application Subarea	UI Element	Property Change
Notes and Tasks	Tasks	Flexible Content: Tasks	Rendered: false
Not applicable	Attachments	Flexible Content: Attachments	Rendered: false

Hide Oracle E-Business Suite UI Elements

There are two major steps required to hide Oracle E-Business Suite UI elements when accessing quotes from Oracle CX Sales:

- Hiding UI Elements
- Hiding the Integration Responsibility

Hiding UI Elements

You use personalization to hide Oracle E-Business Suite UI elements.

Hiding UI Elements on the Quote Creation Page

There are two select buttons to hide on the Quote Creation page. The Customer Name select button and the Contact Name select button. To hide these UI elements on the Quote Creation page do the following:

1. Log in as the Oracle E-Business Suite integration user and navigate to **Integration Responsibility**, **Quote**, and then **Create Quote**.
2. In the **Quote Creation** page, click **Personalize Page**.

   The personalization page organizes all of the controls in a tree structure.
3. Make sure that the **Personalization Level** is set to **Responsibility: Integration Responsibility**.
4. Click the pencil icon beside the **Select** button for Customer Name.
5. Change the **Rendered** property to **false**.
6. Repeat the steps for **Contact Name**.

Hiding UI Elements on the Quote Update Page

UI elements that can be hidden on the Quote Update page using personalization. To hide UI elements on the Quote Update page do the following:

1. Navigate to the **Quote List** page.
2. Click any quote hyperlink to navigate to the **Quote Update** page.
3. Click **Personalize Page**.

   The personalization page organizes all of the controls in a tree structure.
4. Make sure that the **Personalization Level** is set to **Responsibility: Integration Responsibility**.
5. Hide any of the UI elements that you want to hide, using the procedure in Hiding UI Elements on the Quote Creation Page.

Hiding UI Elements on the Add Services Page

You hide UI elements to make the Add Service page read only. To make the Add Services page read only do the following:

1. On the **Product** tab on the **Quote Update** page, select a product.
2. Select **Add Services** from the drop-down list, then click **Go**.
3. Click **Personalize Page**.
4. Make sure that the **Personalization Level** is set to **Responsibility: Integration Responsibility**.
5. Set the **Rendered** property to false for the following UI elements:
   - Select button
   - Multiple Selection: (QotAddSvcSel)

Hiding the Integration Responsibility

After hiding the UI elements, you can hide the integration responsibility. First you must remove the Quotes menu from the integration responsibility. To remove the Quotes Menu Globally from Integration Responsibility do the following:

1. Log in as System Administrator.
2. Navigate to **Function Developer**, **Core Services**, and then **Menus**.
3. Search for the **QOT_QUOTE_ROOT_INT** menu code.

4. Click **Update**.
5. Delete the word **Quotes** from the **Prompts** column.
6. Deselect the **Grant** check box in the same row, then click Apply. The QOT_QUOTE_ROOT_INT menu is removed from Integration Responsibility.

Removing Integration Responsibility from the Navigator

Next you can optionally remove the integration responsibility from the Navigator. To remove Integration Responsibility from the Navigator do the following:

1. Click **Personalize** in the Navigator menu on the home page.
2. In the **Search and Select** area, click **Go**.
3. Select **Integration Responsibility** from the search results, then click **Hide**.
4. Click **Apply**.

Test the Integration

Test the integration by navigating to an Oracle CX Sales opportunity, confirming that the Oracle E-Business Suite quote UI appears, and then clicking Create Quote to confirm that the quote appears in the quote UI.

Configure the Integration with the Security Console

You can use the Security Console to perform unique configurations to your integration, such as restricting access to specific users are particularly administration roles to the Customer Matching UI, and so on.

Perform the following tasks in Oracle CX Sales after you have successfully imported Oracle E-Business Suite items. To create and use a custom role do the following:

1. Click **Navigator**, and select **Security Console**.
2. Click the **Create Role** button, and add the information contained in the following table.

> **Note:** When assigning a role, consider the following: assigning a role to a specific matching user allows individual users who have been assigned that role to perform customer matching. Assigning the role to an existing role hierarchy allows the group of users contained within the role hierarchy to perform customer matching.

Field	Value
Role Name	Customer Matching Admin
Role Code	INT_Customer_Matching_Admin_Role
Role Category	CRM - Job Roles

Field	Value
Description	Custom role for granting the access to perform customer matching.

After you have created the Customer Matching Admin role using Security Console, the role is exposed in Application Composer. You can then assign privileges which allow the role to view custom objects. To assign privileges, see the following procedure.

Defining Security Policies for a Custom Object

To define security policies for a custom object, do the following:

1. Click **Navigator**, and select **Application Composer**.
2. From the **Application** drop-down list, select the relevant application.
3. Expand the Custom Objects node, and then expand the relevant object.
4. Click **Security**, and then in the **Define Policies: Object_name** page, select the appropriate check boxes for the privileges you want the role to have for the object, and then click **Save and Close**.

6 Required Files

Required HTML File: Explained

The following HTML file is required for the integration:

O_INT_EBS_ErrorPage.html

Required Groovy Files

The following Groovy script files are required for global functions used in the integration. Refer to the Related Topics section in this topic for a link to integration implementation files (Doc ID 1645923.1) on My Oracle Support.

The files are located in the Attachments section of the article.

Global Function Groovy Files

The following Groovy script files are required for global functions used in the integration.

- O_INT_GetLogMsg.groovy
- O_INT_GetSysParam.groovy
- O_INT_AddMultiValueCriterialtem.groovy
- O_INT_ApplyFilter.groovy
- O_INT_Debug.groovy
- O_INT_Error.groovy
- O_INT_FindRowByKey.groovy
- O_INT_GetLogMsg.groovy
- O_INT_GetRecordCount.groovy
- O_INT_GetRecords.groovy
- O_INT_EBS_GetSysParam.groovy
- O_INT_Info.groovy
- O_INT_Log.groovy
- O_INT_LogMessage.groovy
- O_INT_Warn.groovy

Integration Configuration Custom Object Groovy Files

The following Groovy script files are required for the Integration Configuration custom object.

Validation Rule

UniqueKey.groovy

Global Functions

- O_INT_GetIntegConfigParameter.groovy
- O_INT_GetIntegConfigParameters.groovy

XREF Custom Object

The following Groovy script files are required for the XREF custom object.

Global Functions

- O_INT_CreateXREF.groovy
- O_INT_GetAllXREF.groovy
- O_INT_GetXREF.groovy
- O_INT_UpdateXREF.groovy

Field Mapping Custom Object

The following Groovy script files are required for the Field Mapping custom object.

Validation Rule

UniqueKey.groovy

Global Function

- O_INT_GetMappedField.groovy
- O_INT_EBS_GetMappedField.groovy

EBS Match Child Object

The following Groovy script files are required for the EBS Match Child object.

Object Function

ForceCreateNewCustomer.groovy

Object Validation

O_INT_EBS_MatchSelectRule.groovy

Triggers

O_INT_CleanupMatchesTrigger.groovy

Account Standard Object

The following Groovy script files are required for the Account standard object.

Global Functions

- O_INT_EBS_CreateCustomer.groovy

- O_INT_EBS_GenerateQuoteURL.groovy
- O_INT_EBS_GetBaseURL.groovy
- O_INT_EBS_SearchCustomer.groovy

Object Functions

- O_INT_EBS_CheckMatch.groovy
- O_INT_EBS_CleanupMatches.groovy
- O_INT_EBS_CustomerQuoteURL.groovy
- O_INT_EBS_CustomerSync.groovy
- O_INT_EBS_GetSearchCriteria.groovy
- O_INT_EBS_PopulateMatches.groovy

Triggers

- O_INT_EBS_CustomerSyncTrigger.groovy
- O_INT_EBS_CustomerCreateTrigger.groovy

Opportunity Standard Object

The following Groovy script files are required for the Opportunity standard object.

Global Functions

- O_INT_EBS_CreateContact.groovy
- O_INT_EBS_IsMatchedContact.groovy
- O_INT_EBS_SearchContact.groovy

Object Functions

- O_INT_EBS_CreateQuote.groovy
- O_INT_EBS_OpptyQuoteURL.groovy
- O_INT_EBS_SearchContactAndCreateXref.groovy
- O_INT_EBS_SyncPrimaryContact.groovy

Triggers

O_INT_CleanUpStatusTrigger.groovy

Related Topics