

Oracle® Fusion Applications CRM Extensibility Guide

11g Release 6 (11.1.6)

Part Number E20388-05

September 2012

Oracle® Fusion Applications CRM Extensibility Guide

Part Number E20388-05

Copyright © 2011-2012, Oracle and/or its affiliates. All rights reserved.

Authors: Kristin Penaskovic, Iqtidar Hussain, Sean Frogner, Steve Muench, Ying Wang, Pundir Prabhakar, Claudio Moraes, Michael Phelan

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

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

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

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

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

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

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

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

Contents

1 Application Composer: Introduction

Extending CRM Applications: Top Tasks	1-1
Extending CRM Applications: How It Works	1-1
Extending CRM Applications Using Sandboxes: Highlights	1-4
Extending CRM Applications Across Application Boundaries: Explained	1-5

2 Application Composer: Using the Application Composer

Defining Objects: Explained	2-1
Object Relationships: Explained	2-6
Defining Fields: Explained	2-11
Field Types	2-13
Defining Pages: Explained	2-36
Creating a Work Area: Explained	2-44
Subtabs: Explained	2-50
Tree Nodes: Explained	2-57
Actions and Links: Explained	2-63
Saved Searches for CRM Objects: Explained	2-67
Securing Custom Objects: Explained	2-70
Groovy Scripting: Explained	2-74
Groovy Scripting: Examples	2-76
Supported Classes and Methods for Use in Groovy Scripts: Explained	2-83
Importing and Exporting Custom Objects: Explained	2-97
FAQs for Extending Oracle Fusion CRM Applications	2-99

3 Application Composer: Extending Oracle Fusion CRM Applications

Customizing Oracle Fusion CRM Applications Using Oracle Fusion CRM Application Composer: Explained	3-1
Oracle Fusion Common CRM Objects: Explained	3-47
Customizing Oracle Fusion Customer Center Pages: Explained	3-54
Customizing Oracle Fusion Sales Forecasting Pages: Explained	3-58

4 Application Composer: Creating Object Workflows

Object Workflows: Explained	4-1
Object Workflows and Oracle Fusion CRM Application Composer: How They Work Together	4-4
Object Workflows: Examples	4-5
Object Workflows and Field Updates: How They Work Together	4-8

Object Workflows and E-Mail Notification: How They Work Together	4-9
Object Workflows and Task Creation: How They Work Together	4-10
Object Workflows and Outbound Message: How They Work Together	4-11
Creating Object-Specific Web Services: Explained	4-13
E-Mail Templates: Explained	4-15
FAQs for Creating Object Workflows	4-17

5 Application Composer: Building Custom Subject Areas

Custom Subject Areas:Explained	5-1
Date Leveling: Explained	5-3
Measures: Explained	5-3
Securing Custom Subject Areas: How It Works	5-4
Extension Dimension Reports: Explained	5-5
Publishing a Custom Subject Area: Explained	5-5
FAQs for Building Custom Subject Areas	5-6

6 Oracle Composer: Customizing Oracle Fusion CRM Applications

Customizing Oracle Fusion CRM Pages Using Oracle Composer: Explained	6-1
Customizing Partner Management Pages Using Oracle Composer: Explained	6-3
FAQs for Customizing Oracle Fusion CRM Applications	6-6

Preface

This Preface introduces the guides, online help, and other information sources available to help you more effectively use Oracle Fusion Applications.

Oracle Fusion Applications Help

You can access Oracle Fusion Applications Help for the current page, section, activity, or task by clicking the help icon. The following figure depicts the help icon.



You can add custom help files to replace or supplement the provided content. Each release update includes new help content to ensure you have access to the latest information. Patching does not affect your custom help content.

Oracle Fusion Applications Guides

Oracle Fusion Applications guides are a structured collection of the help topics, examples, and FAQs from the help system packaged for easy download and offline reference, and sequenced to facilitate learning. You can access the guides from the **Guides** menu in the global area at the top of Oracle Fusion Applications Help pages.

Note

The **Guides** menu also provides access to the business process models on which Oracle Fusion Applications is based.

Guides are designed for specific audiences:

- **User Guides** address the tasks in one or more business processes. They are intended for users who perform these tasks, and managers looking for an overview of the business processes. They are organized by the business process activities and tasks.
- **Implementation Guides** address the tasks required to set up an offering, or selected features of an offering. They are intended for implementors. They are organized to follow the task list sequence of the offerings, as displayed within the Setup and Maintenance work area provided by Oracle Fusion Functional Setup Manager.
- **Concept Guides** explain the key concepts and decisions for a specific area of functionality. They are intended for decision makers, such as chief financial officers, financial analysts, and implementation consultants. They are organized by the logical flow of features and functions.

- **Security Reference Manuals** describe the predefined data that is included in the security reference implementation for one offering. They are intended for implementors, security administrators, and auditors. They are organized by role.

These guides cover specific business processes and offerings. Common areas are addressed in the guides listed in the following table.

Guide	Intended Audience	Purpose
Common User Guide	All users	Explains tasks performed by most users.
Common Implementation Guide	Implementors	Explains tasks within the Define Common Applications Configuration task list, which is included in all offerings.
Information Technology Management, Implement Applications Guide	Implementors	Explains how to use Oracle Fusion Functional Setup Manager to plan, manage, and track your implementation projects, migrate setup data, and validate implementations.
Technical Guides	System administrators, application developers, and technical members of implementation teams	Explain how to install, patch, administer, and customize Oracle Fusion Applications.

For guides that are not available from the Guides menu, go to Oracle Technology Network at <http://www.oracle.com/technetwork/indexes/documentation>.

Other Information Sources

My Oracle Support

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

Use the My Oracle Support Knowledge Browser to find documents for a product area. You can search for release-specific information, such as patches, alerts, white papers, and troubleshooting tips. Other services include health checks, guided lifecycle advice, and direct contact with industry experts through the My Oracle Support Community.

Oracle Enterprise Repository for Oracle Fusion Applications

Oracle Enterprise Repository for Oracle Fusion Applications provides visibility into service-oriented architecture assets to help you manage the lifecycle of your software from planning through implementation, testing, production,

and changes. In Oracle Fusion Applications, you can use the Oracle Enterprise Repository for Oracle Fusion Applications at <http://fusionappsoer.oracle.com> for:

- Technical information about integrating with other applications, including services, operations, composites, events, and integration tables. The classification scheme shows the scenarios in which you use the assets, and includes diagrams, schematics, and links to other technical documentation.
- Publishing other technical information such as reusable components, policies, architecture diagrams, and topology diagrams.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

Comments and Suggestions

Your comments are important to us. We encourage you to send us feedback about Oracle Fusion Applications Help and guides. Please send your suggestions to oracle_fusion_applications_help_ww_grp@oracle.com. You can use the **Send Feedback to Oracle** link in the footer of Oracle Fusion Applications Help.

Application Composer: Introduction

Extending CRM Applications: Top Tasks

The Oracle Fusion CRM Application Composer is but one tool that lets you customize and extend your Oracle Fusion CRM applications. Before you start to extend and customize any application within Oracle Fusion CRM, refer first to the Oracle Fusion Applications Extensibility Guide to learn more about all the extensibility options and tools that are available to you.

The Oracle Fusion Applications Extensibility Guide walks you through the customization process for all Oracle Fusion applications, not just within Oracle Fusion CRM. After reviewing that guide, you can then review the Oracle Fusion CRM Extensibility Guide to understand in more detail how to use the Application Composer to extend and customize an application within Oracle Fusion CRM.

Getting Started: Review the Oracle Fusion Applications Extensibility Guide

- Refer to the Oracle Fusion Applications Extensibility Guide to learn which tools are available to you to change an application. You will also learn about the customization development life cycle, including how to use the Sandbox Manager and Customization Manager.

See: Introduction to Customizing and Extending Oracle Fusion Applications

- The Oracle Fusion Applications Extensibility Guide also includes sections on important customization tasks, such as customizing pages, objects, reports, and security.

See: Business User Customizations and Extensions

- After reviewing the Oracle Fusion Applications Extensibility Guide, you can then review the Oracle Fusion CRM Extensibility Guide, which provides detailed information about using the Application Composer.

Extending CRM Applications: How It Works

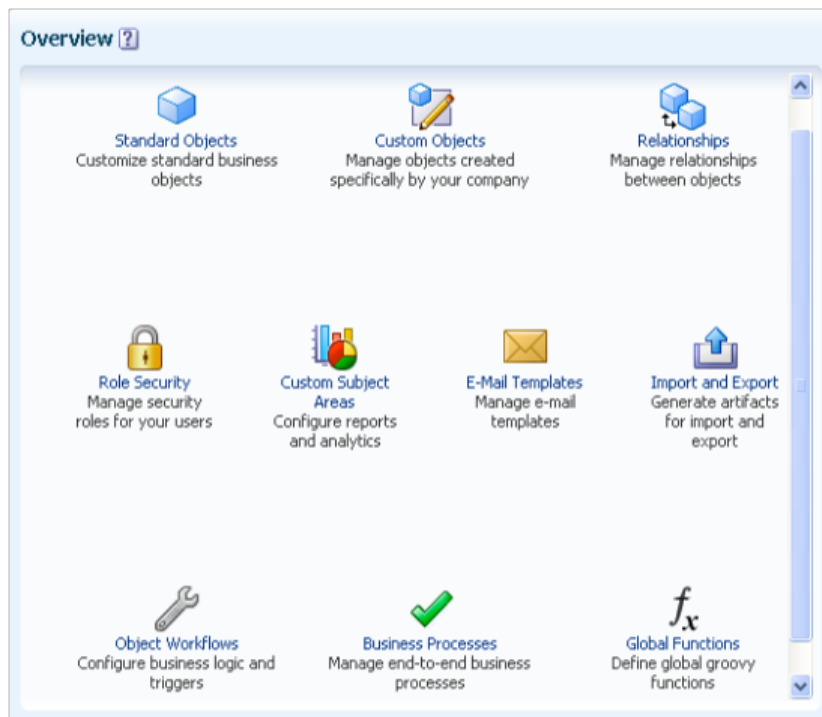
The Oracle Fusion CRM Application Composer is a browser-based configuration tool that enables business analysts and administrators, not just application developers, to customize and extend an Oracle Fusion CRM application. Make the type of data model changes which, for non-CRM applications, can only be

made by application developers. For example, easily create a new object and related fields, then create new Enterprise pages where that object and its fields are exposed to users. The Application Composer is a design time at runtime tool, which means that you can navigate to the Application Composer directly from a CRM application, make your changes, and see most changes take immediate effect in real time, without having to sign back in to the application. Data model changes, such as the creation of custom fields, do require that you reauthenticate before you can see those changes.

Pattern-Based Application Design

The Application Composer hides the complexity of customization from business analysts by leveraging a set of standard design patterns and wizards. You focus on the application changes that your business requires (object model extensions and layout changes, for example), and the Application Composer creates the underlying object artifacts for you.

Access the Application Composer from any CRM application at runtime by using the Navigator menu, and selecting **Application Composer** under the Tools category. The first view of the Application Composer is the main Overview page, which is the entry point into all your customization options.



From the Application Composer's Overview page, you can make application changes such as:

- Customize objects by adding new fields, or create entirely new objects.
- Create foreign key-based relationships between two objects.
- Customize Enterprise pages by exposing your newly created fields for an object, or create an entirely new work area for your newly created objects.

Expose object relationships on Enterprise pages in the form of subtabs or tree nodes.

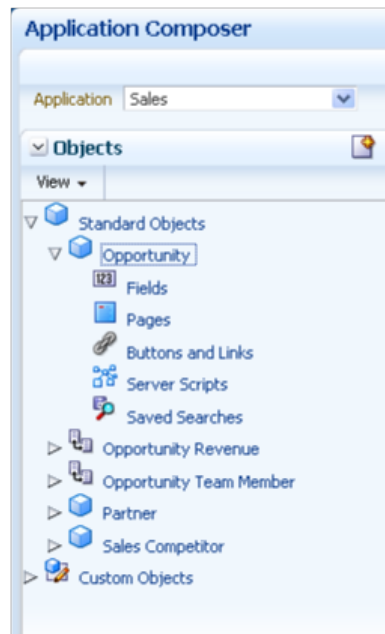
- Write application logic, such as triggers, validation rules, and workflows, for an object or for use across multiple objects.
- Implement functional and data-level security for custom objects.
- Enable objects for custom reporting.

Getting Started: The Application Composer's Overview Page

To access the Application Composer, log in with the **Customer Relationship Management Administrator** job role. Then, select **Application Composer** under the Tools category in the Navigator menu to navigate to the main Overview page.

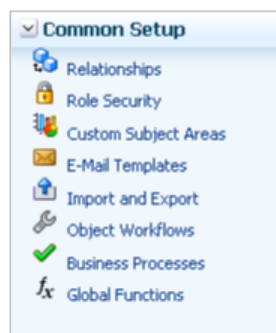
From the main Overview page, select the application you want to customize using the **Application** choice list. Then:

- Use the object tree to select the object you want to customize, or click the New icon to create a new object.



- Use the links in main Overview page, also known as the local area, to select a customization task.

Or, use the links in the Common Setup pane.



Change the selected application in the **Application** choice list at any time to customize another CRM application.

Oracle Fusion Applications Extensibility

The Application Composer is but one tool that lets you customize and extend your Oracle Fusion CRM applications. To learn more about extensibility options that are available to you across all Oracle Fusion applications, see the Oracle Fusion Applications Extensibility Guide.

Extending CRM Applications Using Sandboxes: Highlights

When you customize and extend your Oracle Fusion CRM applications, you should always do so using sandboxes. Sandboxes let users make changes isolated from the mainline application, as well as from other sandboxes. (The mainline is the source of data and definitions used at the time of creating a new sandbox.) Business analysts can implement and test application customizations and, once satisfied, can publish them back to the mainline. When publishing a sandbox, the application customizations that you made in that sandbox overwrite the mainline application's existing configuration.

Before you start to extend and customize any application within Oracle Fusion CRM, refer first to the Oracle Fusion Applications Extensibility Guide to learn more about all the extensibility options and tools that are available to you, including sandboxes. After reviewing that guide, you can then review Customizing Oracle Fusion CRM Applications Using Sandboxes (1487832.1) on My Oracle Support at <https://support.oracle.com>.

Note

Whenever possible, always make your application customizations in sandboxes first, and then publish your changes back to the mainline application. In general, you should not make software changes directly in the mainline unless sandboxes are not available for your particular customization.

Understanding Sandboxes

- Refer to the Oracle Fusion Applications Extensibility Guide to learn which tools are available to you to change an application. You will also learn about the customization development life cycle, including how to use the Sandbox Manager and Customization Manager.

See: Introduction to Customizing and Extending Oracle Fusion Applications

- Refer to Customizing Oracle Fusion CRM Applications Using Sandboxes to learn more about special considerations for using sandboxes when customizing Oracle Fusion CRM Applications. For example, you typically create custom fields for an object in a sandbox. However, you can create custom subject areas for business intelligence reporting only in the mainline application. Therefore, to create a report that includes your custom fields, you must first publish the sandbox in which your custom fields were created. Only then can you create a custom subject area

(in the mainline application) to later use as the basis for any report.
See: Customizing Oracle Fusion CRM Applications Using Sandboxes
(1487832.1) on My Oracle Support at <https://support.oracle.com>.

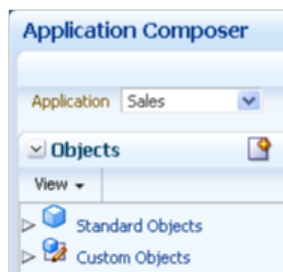
Extending CRM Applications Across Application Boundaries: Explained

When you customize and extend your Oracle Fusion CRM applications, you always do so within the context of a Web application, such as Oracle Fusion Sales or Oracle Fusion Marketing. This is a critical selection that you must remember, because the customizations that you make reside within that application only. For example, to create a new custom object, you first select an application on the main Overview page of the Application Composer. The new custom object will belong only to the application that you select.

Note

When you first open the Application Composer, the default application is always Common. If you previously made customizations in another application, such as Sales, then you must manually change the application using the Application choice list to Sales, before you can review and update those customizations.

Web applications are also referred to as application containers.



The types of changes that you can make in the Application Composer can be categorized into three areas:

- User interface extensions
- Object model extensions
- Scripting extensions

Understanding Web Application Constraints

The concept of application containers directly impacts what you can do when making the above types of changes using the Application Composer.

- User interface extensions
For example, can I add a subtab to an opportunity's page that displays sales lead records?
- Object model extensions

For example, can I create relationships between objects that exist in different Web applications?

- Scripting extensions

For example, for the opportunity object, can I create a trigger whose underlying expression references an object in Oracle Fusion HCM?

The special considerations that you must be aware of when extending and customizing across Web application containers are described in Oracle Fusion Applications CRM Extensibility: Special Considerations (1490074.1) on My Oracle Support at <https://support.oracle.com>.

Application Composer: Using the Application Composer

Defining Objects: Explained

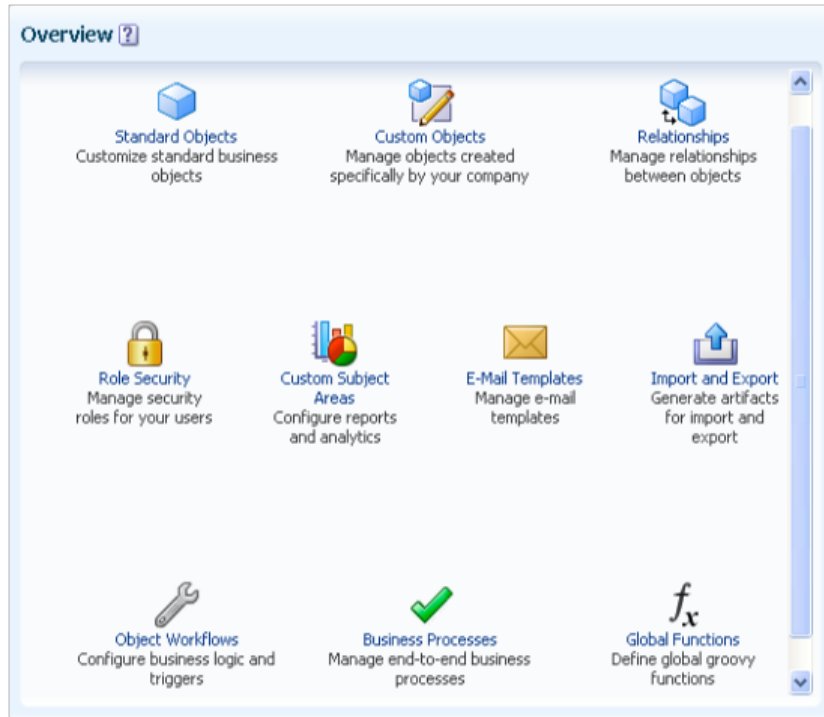
One of the primary customization options available to you when using the Oracle Fusion CRM Application Composer is the ability to extend a CRM application's object model. Customize CRM objects by adding new fields to an existing object (standard objects), or create entirely new objects (custom objects). Standard objects are objects that are delivered with a CRM application, and made available to the Application Composer for customization. Custom objects are objects that you create using the Application Composer. You can create either top-level objects (objects without a parent) or child objects (objects created in the context of a parent).

Review these aspects of the object model extension process in the Application Composer before you begin to customize or extend your CRM application's object model:

- Browsing the object tree
- Creating a custom object
- Using the Object Overview page
- Editing an object's attributes
- Viewing child and related objects
- Deleting a custom object

CRM Application Composer's Object Tree

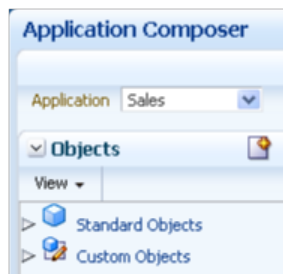
Access the Application Composer from a CRM application at runtime by using the Navigator menu. The first view of the Application Composer is the main Overview page, which is the entry point into all your customization options.



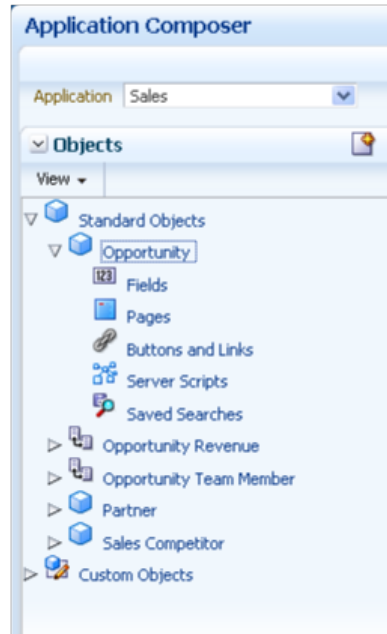
On the main Overview page, the regional pane at left displays the object tree, which lets you browse an application's existing object configuration in a tree format. The object tree reflects the latest configuration of the application: both standard objects as well as custom objects.

To use the object tree:

1. Select **Application Composer** from the Navigator menu, under the Tools category.
2. On the main Overview page, select an application from the **Application** choice list.



3. For each object node, whether standard or custom, expand it further to view and edit object details, such as an object's fields and Enterprise pages where the object is exposed.



Tip

At the top of the object tree, you can also click the New icon to create a new custom object.

For both standard and custom objects, you can view and edit the following details:

- Fields

Add new fields to an object.

- Pages

Modify the pages on which an object appears.

- Buttons and links

Add buttons or links to Enterprise pages.

- Server scripts

Write application logic that controls the behavior of an object's records.

- Saved searches

Define saved searches for an object.

For custom objects, you can also view and edit details for:

- Security

Implement functional and data-level security for an object and its records.

Creating a Custom Object

To create a new custom object, you first select an application on the main Overview page of the Application Composer. The new custom object will belong to the application that you select. After you select the application:

1. Select the **Custom Objects** node or link in either the object tree or local area of the main Overview page.

On the resulting summary table, click the New icon.

2. Or, at the top of the object tree, click the New icon
3. Complete the primary identifying attributes for a custom object:

- a. **Display Label**

An object's display label is the user-friendly label for an object, and also becomes the default page title for the object's work area.

- b. **Plural Label**

The plural label is used when the object is displayed as the detail section of a master-detail page, such as on a subtab.

- c. **Record Name**

Use the **Record Name** field to specify the display label for the object's RecordName field. The RecordName field stores the "name" of the record. For example, for an opportunity object, this RecordName field stores the opportunity's name. Accordingly, if you were creating this object as a custom object, then you would set the **Record Name** field to **Opportunity Name**.

Typically, this field is the object's primary user-recognizable identifier for the object, and as such, is usually the identifier that runtime users drill down on, from the overview page to the detail page.

- d. **Object Name**

The object name is the internal name for the object.

- e. **Description**

Tip

To create a custom child object, click the Create Child Object button in the standard or custom objects summary table, or from an object's Object Overview page.

Once created, a child object cannot be changed to a parent object. Similarly, a parent object cannot be changed to a child object.

Child objects are discussed below.

Using the Object Overview Page

The Object Overview page provides a high-level overview of a standard or custom object. The Object Overview page displays the primary attributes for an object, plus a list of child objects and related objects, if any.



To access the Object Overview page:

1. Select an application on the main Overview page.
2. Select a standard or custom object in the object tree.
3. Or, select the **Standard Objects** or **Custom Objects** node or link in either the object tree or local area of the main Overview page, choose an object from the resulting summary table, and select the Edit icon.

From the Object Overview page, you can:

- Edit the object's primary attributes, described in the previous section.
- View the parent child relationships that were created for this object.
You can also create new child objects from this page, which implicitly creates a new parent child relationship.
- View the non-parent child relationships that were created for this object.

Editing an Object's Attributes

After an object has been created, you can edit its attributes from its Object Overview page.

To edit an object's attributes:

1. Select an application on the main Overview page.
2. Select the **Standard Objects** or **Custom Objects** node or link in either the object tree or local area of the main Overview page.
3. From the resulting summary table, select an object and then select the Edit icon to navigate to its Object Overview page.
4. On the Object Overview page, click **Edit**:
 - Change the object's display label, plural label, description, and record name at any time.
 - You cannot change the Object Name and API Name after the object has been created.

A custom object's API name for an object is automatically derived using the logical name followed by `_c`. You use the API name in Groovy expressions that you build with the expression editor, when writing business logic for the object.

Viewing Child and Related Objects

The Object Overview page displays a list of child objects and related objects, if any, that have been created for an object. You can also create new child objects from this page.

- A child object is an object with a cascade delete relationship to a parent object. This means that if you delete the parent object, then all its children are automatically deleted. A child object does not exist outside the context of the parent object, and does not have its own work area. You cannot change a child object to a parent object after the child object has been created.
- Related objects can exist independently of each other, even if one object is deleted. Related objects are connected in a foreign key-based relationship between two top-level objects, not as parent and child. These types of relationships include reference relationships and dynamic choice list relationships.

Related objects can have either a one-to-many or a many-to-one relationship with the current object. Note that an object can be related to itself to model a hierarchy of the object. In this case, the object itself is displayed on its Object Overview page as a related object. For example, the Department and Sub-department objects would be displayed in this way.

Note

You do not create these types of relationships from this page. Instead, manage relationships from the Relationships page, which you can access from the Application Composer's main Overview page. Or, create a dynamic choice list relationship by creating a dynamic choice list field for an object, which derives its choice list values from another object.

To create a child object for a standard or custom object:

1. Navigate to an object's Object Overview page.
2. Click the **Create Child Object** button. Creating a child object is the same as creating a custom object, except:
 - The current object is automatically displayed as the parent object.
 - Specify the **Child Collection Name** field to specify the internal name for this set of child object records, which can be used later when writing Groovy scripts.

Deleting a Custom Object

The Application Composer does not support the deletion of either standard or custom objects. If you no longer need an object, optionally enter a note in the description that the object is no longer used.

Object Relationships: Explained

A relationship is a foreign key association between two objects, and indicates a connection between two objects' data. You can expose this connection on Enterprise pages through the use of child or related object subtabs or tree nodes. Using the Oracle Fusion CRM Application Composer, you can create one-to-

many relationships between two objects within the same application, where one object's primary identifier is stored in another object's table. A relationship must exist before you can expose the "many" objects on a subtab or tree node that is displayed on the "one" object's details page or tree. For example, an account can have multiple service requests associated to it. To expose a list of service requests associated with a specific account as a subtab on the account's details page, you must first create a one-to-many relationship between the account and service request objects. You can create these relationships implicitly by creating a child object or by creating a dynamic choice list. Or, create relationships explicitly on the Create Relationship page.

Review these aspects of the relationship creation process in the Application Composer before you begin to create relationships between objects:

- Relationship types
- Creating reference relationships
- Adding subtabs or tree nodes
- Many-to-many relationships

Relationship Types

Four types of one-to-many relationships exist:

- Parent child relationship

Parent child relationships are implicitly created when a custom object is created as a child of a top-level object.

When a child object is created, it is created specifically in the context of its parent. A child object does not have its own work area, and the child object is deleted if the parent object is deleted.

View parent child relationships on the parent object's Object Overview page. If a parent child relationship exists, then the child object is listed on the parent's Object Overview page in the Child Objects region. A top-level object can have many child objects.

You can also view the parent child relationship from the child object's Object overview page. If a parent child relationship exists, then the parent object is listed on the child's Object Overview page in the Object Information region. A child object can have only one parent object.

Relationships that are implicitly created from parent child relationships are also viewable on the Relationships page. The relationship name is automatically generated for you.

- Choice list relationship

Choice list relationships are implicitly created between two objects when you create a dynamic choice list field.

View choice list relationships on an object's Object Overview page. If a choice list relationship exists, then the related object is listed on the object's Object Overview page in the Related Objects region.

A dynamic choice list is a field that contains a list of values which are populated from the actual data of another object. For example, you might want to expose on an Enterprise page a dynamic choice list which lets users specify the HR representative of a given department. The HR

Representative choice list is a field that you are adding to the department object, but the list of values is populated by actual employees from the employee object.

In the previous example of making a list of accounts available for selection for a trouble ticket, an account can be tied to multiple trouble tickets. The relationship that is created is a one-to-many relationship between the account and trouble ticket objects, which enables users to store an account identifier in the trouble ticket object's table. In this relationship, the account object is the source object and the trouble ticket object is the target object. If a source object is ever deleted from the system, then at runtime, the dynamic choice list will have no values in it.

Later, you might want to expose a related object subtab on the account details page which shows, for a single account, all the trouble tickets that are related to it. You can create this related object subtab because the relationship was already created when you created the dynamic choice list.

These objects are related objects, not parent child objects; related objects are not deleted if the current object is deleted.

Relationships that are implicitly created from dynamic choice list relationships are also viewable on the Relationships page. The relationship name is automatically generated for you.

Note

Generally, the dynamic choice list that you create results in the implicit creation of a choice list relationship. The exception is if you create a dynamic choice list between a CRM object and a common object: resource, organization contact, organization profile, address. In such cases, relationships are not implicitly created.

- **Reference relationship**

Instead of using a dynamic choice list to implicitly create a relationship between two objects, you can also manually create this relationship as a reference relationship.

Reference relationships are explicitly created between two top-level objects using the Create Relationships page.

View reference relationships on an object's Object Overview page. If a reference relationship exists, then the related object is listed on the object's Object Overview page in the Related Objects region.

Using our previous example, perhaps you don't need to display an HR Representative choice list on a department Enterprise page, but you still want to add a department subtab to an employee's details page. In this case, manually create a reference relationship between the employee and department objects where the employee is the source object and the department is the target object. This enables the creation of the department subtab. Such a reference relationship, however, does not automatically create a corresponding HR Representative choice list for use

on the department Enterprise page. In fact, once you manually create a relationship, you cannot reuse the relationship to create a choice list. This means that you should carefully consider the need for a choice list before you create a reference relationship.

- **Standard relationship**

Standard relationships are relationships that are already created between two standard objects by the Oracle Fusion CRM application.

You can also view standard relationships on an object's Object Overview page. If a standard relationship exists, then the related object is listed on the object's Object Overview page in the Related Objects region.

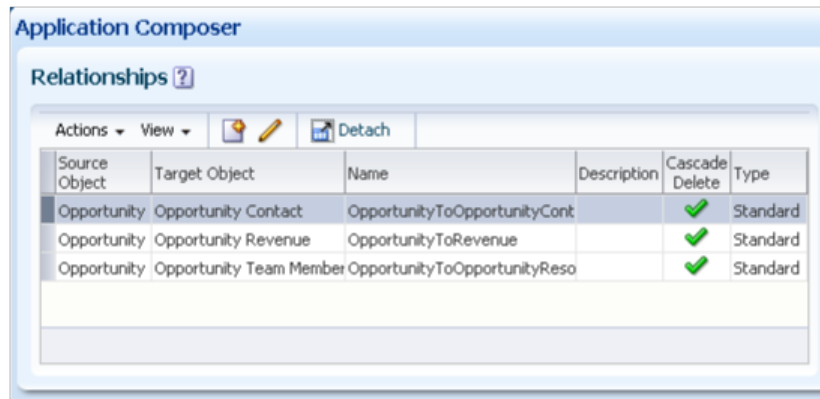
Creating Reference Relationships

Create a foreign key-based, one-to-many relationship between two top-level objects explicitly using the Create Relationship page. Explicitly created relationships are also known as reference relationships.

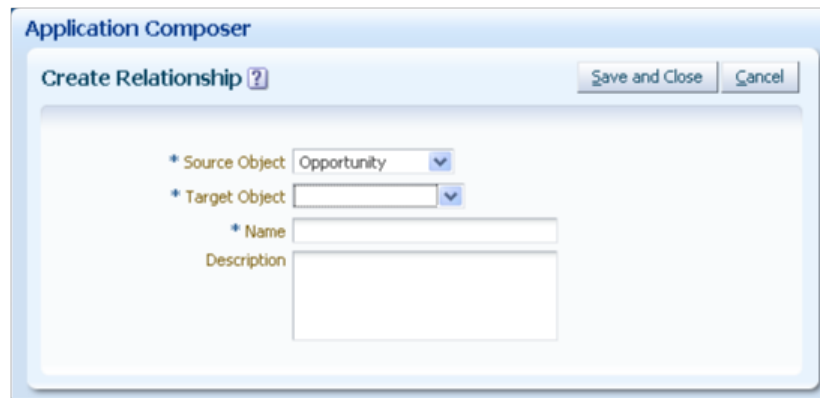
You can also create a foreign key-based, one-to-many relationship by creating child objects and dynamic choice lists. These implicit relationships are discussed in related topics.

To explicitly create a relationship between two top-level objects within the same application:

1. Select **Relationships** in the Common Setup pane.
2. On the Relationships page, click the **New** icon.



3. Select the source object and target object.



A child object cannot be the source object or target object.

Common components, such as notes, interactions, or tasks, are not available for selection as either source objects or target objects.

In general, you create a relationship between two objects within the same application. You can, however, select common objects as target objects. Common objects include:

- Trading Community Resource
- Trading Community Organization Contact
- Trading Community Organization Profile
- Trading Community Address

Once you create a relationship, you can no longer edit the source and target objects.

This relationship adds a field to the target object to store the foreign key details. If the source object is ever deleted, the target object records remain in the system.

4. Enter the relationship name and description.

Once you create a relationship, you can no longer edit the relationship name.

5. Optionally add the target object in a subtab to the source object's detail page, or as a tree node.

Note

You can create multiple relationships between the same source and target objects. For example, create both a Primary Contact and Secondary Contact relationship between the contact and opportunity objects.

Adding Subtabs or Tree Nodes

After you create relationships between objects, you can then expose the "many" objects on a subtab or tree node that is displayed on the "one" object's details page or tree.

When adding a subtab or a tree node to an object's details page or object, you select to add a Child or Related Objects subtab from the object's Pages Overview page. The Application Composer lets you add a subtab or tree node based on any target object that has a relationship with the current object as the source object. Subtabs and tree nodes are discussed in related topics.

Many-to-Many Relationships

Objects can also have a many-to-many relationship. For example, a service request can have multiple employees working on it. At the same time, a single employee can work on multiple service requests. In this scenario, you would create a many-to-many relationship between the service request and employee objects, where the related records from both objects store their primary identifiers in an intersection table.

To create a many-to-many relationship using the Application Composer:

1. Create a child object of one object, and use this child object to represent the intersection table that stores the record identifiers of both objects.

For example, create a service request member object as a child of the service request object. The service request member object's table records the service request identifier as a foreign key.

2. Add a dynamic choice list for the new child object whose related object is the other object in the many-to-many relationship.

For example, create a dynamic choice list, Support Representative, for the service request member object where the choice list's related object is employee. The Application Composer automatically creates the underlying relationship for you, where the employee is the source object and the service request member is the target object. The service request member object's table records the employee identifier as a foreign key.

Now, the service request member object's table records two foreign keys: one for the service request object and the other for the employee object. This enables the many-to-many relationship. You can now do the following:

- Create a child subtab on a service request's details page. The subtab displays all employees that are working on a specific service request.
- Create a related object subtab on an employee's details page. The subtab displays all service requests that an employee is working on, since each employee can work on multiple service requests.

Defining Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. A standard object has a set of standard fields. Standard fields are the fields that are delivered for a standard object in a CRM application. The fields that you add to an object are custom fields. When creating a custom field, the Application Composer provides a set of field types that you can choose from. For example, you can create a check box field, or create a long text field. When you create custom fields for objects and expose the fields on Enterprise pages, the Application Composer automatically creates all the underlying object artifacts for you, and provides full Web service support for those new fields, as well. The Application Composer also makes it easy to enable your object model extensions for importing and exporting.

Review these aspects of editing fields in the Application Composer before you begin to customize or extend your CRM application's object model:

- Adding fields to objects
- Deleting fields

Adding Fields to Objects

Use the Fields page to review the list of standard and custom fields for an object, and create custom fields. A CRM object can have a maximum of 625 fields.

To view the Fields page for an object:

1. Select an application from the **Application** choice list on the main Overview page.
2. Select either the **Standard Objects** or **Custom Objects** node in the object tree to expand the list of objects.
3. Select the object itself to further expand the tree hierarchy.
4. Select the Fields node to navigate to the Fields page.

On the Fields page:

- **Standard Fields**

Review the list of standard fields that are delivered for an object, and optionally modify the display label and help text for a field.

The list of standard fields includes all the fields that are specific to an object, as well as system fields, which could include:

- CreatedBy
- CreationDate
- Id
- LastUpdateDate
- LastUpdatedBy
- RecordName

The custom objects that you create also contain these same system fields, among others.

- **Custom Fields**

Review the list of custom fields that were created specifically for your CRM implementation for either standard or custom objects, and create new custom fields.

To create a custom field, select the New icon from the custom fields table on the Fields page. The Application Composer provides a set of field types that you can choose from when creating new fields:

- Text
- Long text
- Number
- Date
- Datetime
- Check box
- Percentage
- Currency

- Formula
- Fixed choice list
- Dynamic choice list

Deleting Fields

The Application Composer does not support the deletion of either standard or custom fields from objects. If you no longer need a field, optionally enter a note in the field description that the field is no longer used.

Field Types

Field Types and Field Properties: Explained

When you create a custom field, you select from a set of standard field types. Each field type has a corresponding set of standard properties. Some properties are unique to a specific field type, whereas other properties are common across field types. For example, for all field types, you must specify a display label for the field to indicate how you want the field to appear on an Enterprise page.

Before you create a new field for a object, you should understand:

- The set of standard field types available for field creation
- The common set of field properties that you must specify for a field
- How field types work with other components

Field Types

When creating a new field for a object, the Oracle Fusion CRM Application Composer provides a set of standard field types that you can choose from.

The types of fields that you can create are listed below.

- **Text**
Users can enter a combination of letters, numbers, or symbols. This field type is limited to 254 characters.
- **Long text**
Users can enter a combination of letters, numbers, or symbols. This field type supports 32,000 characters.
- **Number**
Users can enter a number in this field.
- **Date**
Users can enter a date, or select a date from a calendar.
- **Datetime**
Users can enter a date, or select a date from a calendar, and enter a time of day. During field creation, you choose whether to show the date or time, or both.
- **Check box**
Users can select a check box, indicating a true or false attribute of a record.
- **Percentage**

Users can enter a percentage. The system automatically adds the percent sign.

- **Currency**

Users can enter a currency amount.

- **Fixed choice list**

Users can select from a list of static values, populated from an FND lookup type.

- **Dynamic choice list**

Users can select from a list of values, populated from another object's set of records.

- **Formula**

A formula field is a field that is calculated in the runtime CRM application using the Groovy-based expression included in the formula field's definition. This is a read-only field that users in the runtime application do not update. However, the application logic that you write can update these fields directly.

Common Field Properties

When you create a custom field, you first select the field type. For example, are you creating a check box field, a formula field, or a long text field? You cannot change the field type after the field is created. The specified field type controls which field properties you must define when creating the field. Some properties are common across field types, whereas other properties are unique to a specific field type.

The common field properties that you can define for a custom field are listed in this table, along with the regions on the field configuration pages where they appear and a list of the applicable field types that you must set these properties for. Use this table to understand the common properties that you must define when creating a new field.

Field Property	Field Property Region	Related Field Types
Label Specify the display label for the field. A maximum length of 80 characters is recommended, although no maximum length is actually enforced.	Appearance These properties control how the field appears to your users in the runtime application.	Set this property for all field types.
Help Text The help text displays when users hover over the field in the runtime application. A maximum length of 80 characters is recommended, although no maximum length is actually enforced.	Appearance	Set this property for all field types.

<p>Display Width</p> <p>Specify the display width for most field types at runtime. The display width is the actual character width for the field on an Enterprise page.</p> <hr/> <p>Tip</p> <p>When setting the display width, consider the resolution in use where this field will be displayed on an Enterprise page. A display width that is too wide will stretch beyond the resolution of the display and result in scroll bars.</p> <p>Generally, enter a display width of no more than 20 to 25 characters.</p> <hr/>	<p>Appearance</p>	<p>Set this property for all field types except for check box, date, and datetime fields.</p>
<p>Name</p> <p>Enter a unique field name, which is for internal use only.</p> <p>The field name is automatically populated based on the field label you enter, but without spaces.</p> <p>Field names can contain only underscores and alphanumeric characters. They must begin with a letter, not include spaces, not end with an underscore, and not contain consecutive underscores. Field names are limited to 28 characters.</p> <p>You cannot change this property after the field is created.</p> <hr/> <p>Tip</p> <p>It is possible to create custom fields with different names, but the same display label. Avoid this scenario, however, so that you do not see two fields with the same display label when configuring an Enterprise page.</p> <hr/> <p>Note</p> <p>The API name is also automatically generated for a field, by taking the logical name and appending <code>_c</code>. The API name is used in your Groovy scripts.</p> <hr/>	<p>Name</p>	<p>Set this property for all field types.</p>

<p>Description</p> <p>Enter a unique field description, which is for internal use only.</p>	<p>Name</p>	<p>Set this property for all field types.</p>
<p>Required</p> <p>Indicate if the field is a required field. You can also optionally use the expression editor to write an expression that describes the conditions required for this field to be required.</p> <hr/> <p>Note</p> <p>If you write an expression to control whether a field is required, then you must also configure the Depends On choice list. This choice list includes fields from the current object, and is used in the evaluation of your expression at runtime.</p>	<p>Constraints</p> <p>Specify constraints, which let you control the runtime behavior of the field.</p>	<p>Set this property for all field types except for formula fields.</p>
<p>Updateable</p> <p>Indicate if the field is an updateable field. You can also optionally use the expression editor to write an expression that describes the conditions required for this field to be updateable. This includes being updateable on an Enterprise page, via Web services, through the import and export functionality, and by server scripts.</p> <hr/> <p>Note</p> <p>If you write an expression to control whether a field is updateable, then you must also configure the Depends On choice list. This choice list includes fields from the current object, and is used in the evaluation of your expression at runtime.</p>	<p>Constraints</p>	<p>Set this property for all field types except for formula fields.</p>
<p>Searchable</p> <p>Indicate if you want this field to be made available for selection as an additional search criteria from the Add Fields choice list in the Advanced Search mode.</p>	<p>Constraints</p>	<p>Set this property for all field types except for long text and formula fields.</p>

<p>Indexed</p> <p>Enable faster searching by indexing this column.</p> <p>Only a limited number of columns are indexed. Accordingly, use this property only on the most frequently searched fields.</p> <p>You cannot change this property after the field is created.</p>	<p>Constraints</p>	<p>Set this property for text, number, date, datetime, currency, and percentage field types.</p> <p>You cannot index long text, formula, and check box fields, or fixed and dynamic choice lists.</p> <hr/> <p>Note</p> <p>You cannot index long text fields. Instead, your users can use the Oracle Fusion Applications search capability to search these field types.</p>
<p>Fixed Value</p> <p>Specify a literal default value for the field.</p> <hr/> <p>Warning</p> <p>Do not assign a literal default value to fields that are both required and intended to be unique, as a runtime error could occur.</p>	<p>Default Value</p>	<p>Set this property for all field types except for formula fields and dynamic choice lists.</p>
<p>Expression</p> <p>Use the expression editor to write an expression that dynamically sets the default value for a field at runtime.</p>	<p>Default Value</p>	<p>Set this property for all field types except for check box and formula fields, and fixed and dynamic choice lists. To set default values for these types of fields, write server scripts.</p>

How Fields Types Work With Other Components

When you create new fields for objects, the Application Composer limits you to a set of standard field types. The field types that you can select from are already integrated with other components of the CRM Extensibility Framework to provide you with the most flexibility possible when customizing and extending your CRM implementation:

- All field types correspond to API data types; each field type has an API name, such as customfield_c.

When writing a server script using the expression editor, use this `_c` field name to reference fields.

- All field types correspond to your Web service XSD payload.
- All field types correspond to your import ODI mappings when using the Application Composer's import and export feature.
- Most field types correspond to available fields that you can use to create a custom subject area for reporting. Exceptions include long text, check box, and formula fields.

Text Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a text field. A text field is a field where users in the runtime application can enter a combination of letters, numbers, or symbols.

Text Field Properties

Create a text field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the text field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints
Indexed	Constraints
Fixed Value	Default Value
Expression	Default Value

The following properties are unique to only certain field types, including text fields:

Field Property	Field Property Region
Display Type Indicate if you want this text field to render in the runtime application as a simple text box. Or, indicate if the field should allow multiple lines where text can wrap, or where the user can enter carriage returns.	Appearance
Maximum Length Indicate the maximum number of characters that a user can enter in the field. You can set a maximum length of 254 characters. If the field is a multiline field, then carriage returns are permitted and count as characters against the total.	Constraints
Minimum Length Indicate the minimum number of characters that a user can enter into the field.	Constraints

Additional Text Field Specifications

Additional specifications for this field type include the following details:

- Data type is VARCHAR2 (254 char).
- A object can have a total of 625 fields. Of those 625 fields, 350 fields are reserved for text and check box fields, and fixed and dynamic choice lists.

Long Text Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a long text field. A long text field is a field where users in the runtime application can enter a combination of letters, numbers, or symbols. This field type supports 32,000 characters.

Long Text Field Properties

Create a long text field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the long text field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Fixed Value	Default Value
Expression	Default Value

The following properties are unique to only certain field types, including long text fields:

Field Property	Field Property Region
Display Type Indicate if you want this text field to render in the runtime application as a simple text box. Or, indicate if the field should allow multiple lines where text can wrap, or where the user can enter carriage returns.	Appearance

Additional Long Text Field Specifications

Additional specifications for this field type include the following details:

- Data type is CLOB.
- A object can have a total of 625 fields. Of those 625 fields, 25 fields are reserved for long text fields.
- The long text field type is not supported by custom subject areas. This means that you cannot add long text fields to a custom report.

Number Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a number field. A number field is a field where users in the runtime application can enter a number.

Number Field Properties

Create a number field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the number field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints
Indexed	Constraints
Fixed Value	Default Value
Expression	Default Value

The following properties are unique to only certain field types, including number fields:

Field Property	Field Property Region
Decimal Places Specify how many numbers can be entered and displayed to the right of the decimal point. If at runtime, a user enters more numbers after the decimal point, then the Application Composer rounds up (using the tie-breaking rule, round half up) to derive the field's value. For example, if you enter 2 for the number of decimal places, then at runtime, an entry of 4.986 is displayed as 4.99.	Constraints

<p>Maximum Length</p> <p>Specify how many numbers a user can enter in the field.</p> <p>During field creation, consider how this property interacts with these other field properties:</p> <ul style="list-style-type: none"> • Display Width <p>If you set a maximum length that is longer than the display width, then users will have to scroll inside the field at runtime to see the number in this field.</p> <ul style="list-style-type: none"> • Decimal Places <p>Maximum Length - Decimal Places = the number of digits that can appear to the left of the decimal point.</p> <p>Do not set a maximum length that is shorter than the number of decimal places.</p>	Constraints
<p>Minimum Value</p> <p>Indicate the minimum numerical value that a user can enter into this field.</p>	Constraints
<p>Maximum Value</p> <p>Indicate the maximum numerical value that a user can enter into this field.</p>	Constraints

Additional Number Field Specifications

Additional specifications for this field type include the following details:

- Data type is NUMBER.
- A object can have a total of 625 fields. Of those 625 fields, 200 fields are reserved for number, currency, and percentage fields.
- Leading zeros are removed.

Date Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a date field. A date field is a field where users in the runtime application can enter a date, or select a date from a calendar. This type of field has no time component.

Date Field Properties

Create a date field by specifying values for the common set of field properties, such as display label and field name.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints
Indexed	Constraints
Fixed Value	Default Value
Expression	Default Value

Additional Date Field Specifications

Additional specifications for this field type include the following details:

- Data type is TIMESTAMP.
- A object can have a total of 625 fields. Of those 625 fields, 50 fields are reserved for date and datetime fields.
- When you create a custom subject area to be used for custom reporting, you can select fields with this type to use for date leveling.

Datetime Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a datetime field. A datetime field is a field where users in the runtime application can enter a date, or select a date from a calendar, and enter a time of day. You can show the date or time, or both.

Datetime Field Properties

Create a datetime field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the datetime field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Name	Name
Description	Name
Required	Constraints

Updateable	Constraints
Searchable	Constraints
Indexed	Constraints
Fixed Value	Default Value
Expression	Default Value

The following property is unique to only certain field types, including datetime fields:

Field Property	Field Property Region
Display Type Indicate if you want this datetime field to show the date or time, or both.	Appearance

Additional Datetime Field Specifications

Additional specifications for this field type include the following details:

- Data type is TIMESTAMP.
- A object can have a total of 625 fields. Of those 625 fields, 50 fields are reserved for date and datetime fields.
- When you create a custom subject area to be used for custom reporting, you can select fields with this type to use for date leveling.
- This field type supports time zone conversion.

Check Box Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a check box field. A check box field is a field where users in the runtime application can select a check box, indicating a true or false attribute of a record.

Check Box Field Properties

Create a check box field by specifying values for the common set of field properties, such as display label and field name.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Name	Name

Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints
Fixed Value	Default Value

Additional Check Box Field Specifications

Additional specifications for this field type include the following details:

- Data type is VARCHAR2.
- A object can have a total of 625 fields. Of those 625 fields, 350 fields are reserved for text and check box fields, and fixed and dynamic choice lists.
- The check box field type is not supported by custom subject areas. This means that you cannot add check box fields to a custom report.

Percentage Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a percentage field. A percentage field is a field where users in the runtime application can enter a percentage. The Application Composer automatically adds the percent sign to the number.

Percentage Field Properties

Create a percentage field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the percentage field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints
Indexed	Constraints
Fixed Value	Default Value
Expression	Default Value

The following properties are unique to only certain field types, including percentage fields:

Field Property	Field Property Region
<p>Decimal Places</p> <p>Specify how many numbers can be entered and displayed to the right of the decimal point. If at runtime, a user enters more numbers after the decimal point, then the Application Composer rounds up (using the tie-breaking rule, round half up) to derive the field's value.</p> <p>For example, if you enter 2 for the number of decimal places, then at runtime, an entry of 4.986 is displayed as 4.99.</p>	Constraints
<p>Maximum Length</p> <p>Specify how many numbers a user can enter in the field.</p> <p>During field creation, consider how this property interacts with these other field properties:</p> <ul style="list-style-type: none"> • Display Width <p>If you set a maximum length that is longer than the display width, then users will have to scroll inside the field at runtime to see the amount in this field.</p> <ul style="list-style-type: none"> • Decimal Places <p>Maximum Length - Decimal Places = the number of digits that can appear to the left of the decimal point.</p> <p>Do not set a maximum length that is shorter than the number of decimal places.</p>	Constraints

Additional Percentage Field Specifications

Additional specifications for this field type include the following details:

- Data type is NUMBER.
- A object can have a total of 625 fields. Of those 625 fields, 200 fields are reserved for number, currency, and percentage fields.
- The Application Composer automatically adds the percent sign.

Currency Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a

currency field. A currency field is a field where users in the runtime application can enter a currency amount.

Currency Field Properties

Create a currency field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the currency field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints
Indexed	Constraints
Fixed Value	Default Value
Expression	Default Value

The following properties are unique to only certain field types, including currency fields:

Field Property	Field Property Region
Minimum Value Indicate the minimum numerical value that a user can enter into this field.	Constraints
Maximum Value Indicate the maximum numerical value that a user can enter into this field.	Constraints
Exchange Date Optionally specify the exchange date to use to calculate the currency conversion rate. <hr/> Tip To use the system date when the record was created as the exchange date, specify the field's creation date as the exchange date.	Exchange Date

Additional Currency Field Specifications

Additional specifications for this field type include the following details:

- Data type is NUMBER.
- A object can have a total of 625 fields. Of those 625 fields, 200 fields are reserved for number, currency, and percentage fields.

Note

Each currency field uses two number type columns: one column stores the amount itself and the other column stores the currency conversion rate that is calculated from the entered amount's currency code to the corporate currency code.

- A CRM object includes the following fields to assist with currency conversion. These fields are automatically added to a CRM object, provided that the object's CRM application allows the creation of currency fields, and are derived from the CRM application's corporate currency setup.

- Currency code

This is the currency code for all currency fields for an object.

- Corporate currency code
- Currency conversion rate type

Currency conversion for a currency field occurs as follows:

- At runtime, the user enters the currency amount.
- When the user saves the record:
 - The currency amount is stored using the currency code specified for the object.
 - The CRM application calculates the currency conversion rate using the object's currency code, corporate currency code, currency conversion rate type, and the currency field's specified exchange date, if any.

In addition to the entered amount, only the conversion rate that is calculated from the entered amount's currency code to the corporate currency code is stored.

- If you later change either the currency code or exchange date, the CRM application recalculates the currency conversion rate for the record.

Fixed Choice Lists: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a fixed choice list. A fixed choice list is a field that contains a list of static values which

are populated from FND lookup types. At runtime, your users can select one or more values from this field, depending on the field's definition.

Fixed Choice List Properties

Create a fixed choice list by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the fixed choice list field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints
Fixed Value	Default Value
Tip If this choice list allows multiple values, you can still write an expression to preselect multiple values by default. For example, if the field is comprised of three lookup codes with (Code,Label) of (S,Small), (M,Medium),(L,Large),(XL,Extra Large), then to preselect the Small and Extra Large selections by default, set the default value to the literal string (without quotes): S,XL .	

The following properties are unique to only certain field types, including fixed choice lists:

Field Property	Field Property Region
Display Type Indicate if your users can select only one value, or multiple values, from the choice list at runtime. Selecting the display type is possible only during field creation.	Appearance
Lookup Type	List of Values
Constrain List by Parent Field Value Selection	List of Values

Using the List of Values Region

The values in a fixed choice list are populated from FND lookup types. Select the lookup type whose values you want to display in this choice list. Selecting the lookup type is possible only during field creation.

Or, create a new lookup type and add new values to it. You can also enter a lookup type and select the Edit icon to modify the existing values.

The set of FND lookup types that are available for selection is constrained to those lookup types that are related to this fixed choice list's object (via product code), as well as all custom lookup types that have been created for your CRM implementation.

The 'List of Values' window is titled 'List of Values' with a checkmark icon. It contains the following elements:

- Instructional text: 'Configure the list of values you want to display in the choice list. Click the search icon to select a predefined lookup type, or create a new one.'
- '* Lookup Type' field: A text box containing 'TT Area' with search, list, and edit icons to its right.
- 'Constrain list by parent field value selection' checkbox: A checked checkbox.
- Instructional text: 'Select the parent field whose value selection will drive the contents of this field, and then define the mapping between the parent and child values.'
- 'Parent Choice List' dropdown: A dropdown menu showing 'Type'.
- 'Parent Lookup Type' text: 'TT Type'.
- '* Value Map' button: A button with a pencil icon.

You can constrain the actual values that display in this fixed choice list at runtime by relating this fixed choice list to a parent fixed choice list. The value selected in the parent fixed choice list drives the values that display in this fixed choice list. For example, you might want your users to see two choice lists on an Enterprise page where they can create a trouble ticket: one choice list for specifying the trouble ticket type and one choice list for specifying the trouble ticket area. If a user selects **Hardware** from the **Type** choice list, then the **Area** choice list should contain a list of only hardware options against which the trouble ticket can be logged.

To do this, select the **Constrain List by Parent Field Value Selection** check box, select the parent field, and then map the values between the parent lookup type and this field's lookup type.

The 'Edit Value Map' window is titled 'Edit Value Map'. It contains the following elements:

- Instructional text: 'Select a parent lookup value, and then select the child lookup values to display for it.'
- 'Parent Lookup Type' text: 'TT Type'.
- 'Child Lookup Type' text: 'TT Area'.
- 'TT Type' list: A list box containing 'Hardware', 'Network', and 'Software'. 'Hardware' is selected.
- 'TT Area: Hardware' section: Contains two list boxes and navigation buttons.
 - 'Available Lookups' list: Contains 'Application', 'Operating System', and 'Router'.
 - 'Selected Lookups' list: Contains 'CD-ROM', 'Desktop', and 'Laptop'.
 - Navigation buttons: A set of four arrows (right, double right, left, double left) between the two list boxes.
- 'OK' and 'Cancel' buttons: Located at the bottom right.

Note

The **Constrain List by Parent Field Value Selection** check box is available for selection only during field creation, and only if at least one other fixed choice list, which is a single-select choice list, has been defined.

After field creation, however, you can update the mapping between lookup values.

To implement the previous example:

1. Define the **Type** fixed choice list.
2. Define the **Area** fixed choice list.
3. Select the **Constrain List by Parent Field Value Selection** check box and select the parent field, **Type**.
4. Finally, map the values between the Type and Area lookup types.

For example, map all relevant hardware values in the Area lookup type's set of values, such as desktop and laptop, to the value of Hardware in the Type's lookup type.

Additional Fixed Choice List Specifications

Additional specifications for this field type include the following details:

- Data type is VARCHAR2 (1500).

Dynamic Choice Lists: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a dynamic choice list. A dynamic choice list is a field that contains a list of values which are populated from the actual data of another object. For example, you might want to expose on an Enterprise page a dynamic choice list which lets users specify the account that they are logging a trouble ticket against. In this example, the **Account Name** choice list is a field that you are adding to the trouble ticket object, but the list of values is populated by actual names from the account object.

When creating dynamic choice lists, review the following:

- Review the common set of field properties, as well as the dynamic choice list-specific properties, that you must specify.
- Review the options available in the List Data Source, Additional List Display Values, and Additional List Search Fields regions.
- Understand how a dynamic choice list results in the implicit creation of a relationship.

Note

When you are ready to add this dynamic choice list to a page, note that you cannot add dynamic choice lists to the local search region of a custom work area.

Dynamic Choice List Properties

Create a dynamic choice list by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the dynamic choice list field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name
Required	Constraints
Updateable	Constraints
Searchable	Constraints

The following properties are unique to only certain field types, including dynamic choice lists:

Field Property	Field Property Region
Related Object	List Data Source
List Selection Display Value	List Data Source
Data Filter	List Data Source
Additional List Display Values	Additional List Display Values
Additional List Search Fields	Additional List Search Fields

Using the List Data Source, Additional List Display Values, and Additional List Search Fields Regions

When defining a dynamic choice list, use the following regions to determine what data will display in the list of values at runtime.

- List Data Source region

List Data Source

Select the related object you want to query to populate the list. Then, select the related object field value you want to display for the user's selection.

Related Object: Account

List Selection Display Value: Name

Data Filter

Add search field criteria to filter the query of the related object specified above.

Add Search Field...

No search fields added.

- Related Object

The values in a dynamic choice list are populated from another object's data. Select the related object first, then use the **List Selection Display Value** choice list to select the related object's field that you want to

expose as a field for your own object. Selecting the related object is possible only during field creation.

Note

The set of objects that are available for selection is constrained to top-level objects only. You cannot select a child object as a related object.

In our example, the related object would be **Account**.

Tip

Once you create a dynamic choice list, you can easily recognize the choice list's related object from the Fields page. The Fields page displays summaries of both standard and custom fields for the selected object. If a dynamic choice list was created, then the Type column includes the related object. In our example, the field type would be **Choice List (Dynamic) <Account>**.

- List Selection Display Value

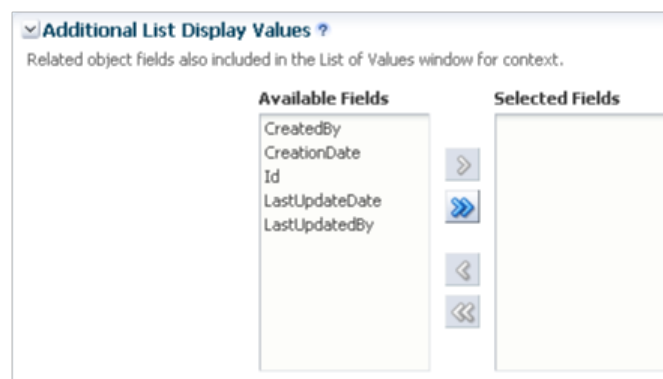
The **List Selection Display Value** choice list is the related object's field that is displayed within the dynamic choice list as the first column at runtime. This is the primary field on the related object that your users will use to make the appropriate selection. In our example, the field might be something like **Name**.

- Data Filter

You can further refine the set of data that appears within the dynamic choice list at runtime by using data filters.

In our example, we could filter out any accounts outside a particular region.

- Additional List Display Values region



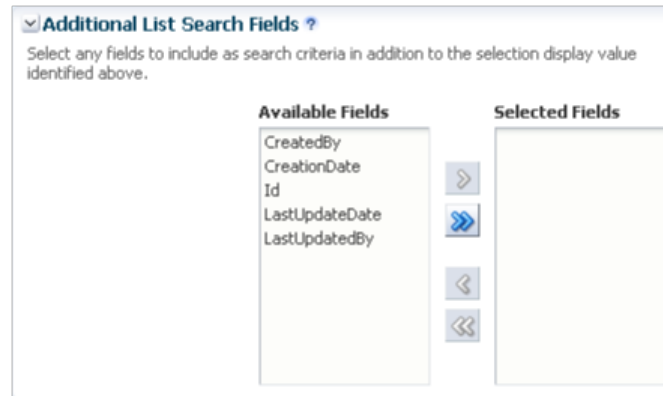
You can further refine the look and feel of the dynamic choice list by selecting additional fields to display in the choice list.

Use the **Additional List Display Values** shuttle to include additional related object fields in the dynamic choice list at runtime. These additional

fields assist your users in making a selection from the choice list. The shuttle does not include the field that you already selected in the **List Selection Display Value** choice list.

There is no limit on the number of additional fields that you can select.

- Additional List Search Fields region



You can indicate which additional related object fields can be added as search criteria in the dynamic choice list's Search and Select dialog.

Use the **Additional List Search Values** shuttle to include additional related object fields in the dynamic choice list's Search and Select dialog, accessed using the **Search...** link at runtime. The shuttle does not include the field that you already selected in the **List Selection Display Value** choice list.

There is no limit on the number of additional fields that you can select.

Implicit Relationship Creation

When you create a dynamic choice list for an object based on a related object, you are implicitly creating a one-to-many foreign key relationship where the current object is the "many" object and the related object is the "one" object. This implicit creation of a relationship lets you later add a related object subtab for the "many" object on the "one" object's details page. You can view these implicitly created choice list relationships on the Relationships page.

In the previous example of making a list of accounts available for selection for a trouble ticket, an account can be tied to multiple trouble tickets. The relationship that is created is a one-to-many relationship between the account and trouble ticket objects, which enables users to store an account identifier in the trouble ticket object's table. In this relationship, the account object is the source object and the trouble ticket object is the target object. If a source object is ever deleted from the system, then at runtime, the dynamic choice list will have no values in it.

Later, you might want to expose a related object subtab on the account details page which shows, for a single account, all the trouble tickets that are related to it. You can create this related object subtab because the relationship was already created when you created the dynamic choice list.

Note

Generally, the dynamic choice list that you create results in the implicit creation of a choice list relationship. The exception is if you create a dynamic choice list

between a CRM object and a common object: resource, organization contact, organization profile, address. In such cases, relationships are not implicitly created.

Additional Dynamic Choice List Specifications

Additional specifications for this field type include the following details:

- Data type is VARCHAR2 (1500).

Formula Fields: Explained

Using the Oracle Fusion CRM Application Composer, you can extend a CRM application's object model by adding new fields to both standard or custom objects. One field type that you can add to a custom or standard object is a formula field. A formula field is a field that is calculated in the runtime CRM application using the Groovy-based expression included in the formula field's definition. For example, write an expression to calculate an employee's annual bonus amount.

Formula Field Properties

Create a formula field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the formula field type.

The following properties are common across multiple field types:

Field Property	Field Property Region
Label	Appearance
Help Text	Appearance
Display Width	Appearance
Name	Name
Description	Name


The following properties are unique to only certain field types, including formula fields:

Field Property	Field Property Region
Formula Type Specify the field's data type, such as text, number, or date. The type can be specified only during field creation.	Field Value Type
Display Type If the formula type is Text , then indicate if you want this formula field to render in the runtime application as a simple text box. Or, indicate if the field should allow multiple lines where text can wrap.	Appearance
Depends On	Constraints

Using the Expression Editor and the Depends On Choice List

Use the **Depends On** choice list to indicate if the field should be automatically recalculated (using the expression you write) if another field's value changes.

Constraints

☐ Updateable  Depends On: Contact Last Name

Minimum Length: Characters

Maximum Length: Characters

Note

The **Depends On** choice list includes a list of fields that belong to the same object. If you want this formula field to automatically recalculate if the value of another field on a different object changes, then you must write a server script.

Use the expression editor to write an expression that will calculate the field's value at runtime.

Expression Builder

Hide Palette

Functions Fields Keywords

Category	Function	Description
Text	contains	
Text	find	
Text	endsWith	
Text	left	
Text	length	
Text	right	
Text	startsWith	
Text	upperCase	
Text	lowerCase	
Text	substringBefore	
Text	substringAfter	

Insert

OK Cancel

For example, if your expression calculates the value of an employee's annual bonus amount, then you could set the expression to automatically recalculate that amount if the employee's salary changes. Whenever the salary changes, the bonus field immediately reflects the new bonus amount without your users having to refresh the employee's record.

In another example, if your expression determines the right customer phone number to use for an opportunity, then you could set the expression to automatically reset the phone number if the opportunity's customer account changes. Whenever the customer account changes, the phone number field immediately reflects the new phone number without your users having to refresh the opportunity record.

Additional Formula Field Specifications

Additional specifications for this field type include the following details:

- Data type is set by the **Formula Type** property.
- The formula field type is not supported by custom subject areas. This means that you cannot add formula fields to a custom report.
- You cannot search on a formula field.

Defining Pages:Explained

You can expose web pages in web applications, mobile applications, and tablets. Objects are of two types: standard and custom. After you extend a CRM application's object model, your next step is to expose those new objects and fields on the Enterprise Pages tab. Customizing and creating enterprise pages in CRM is a simplified process because the pages available to display an object are limited to a set of page types. Every top-level CRM object has an overview page, a creation page, and a details page, collectively known as a work area. When you create new fields, the Oracle Fusion CRM Application Composer provides work area configuration pages where you can add those fields for display. Or, you can create a new work area for a custom object and its fields by using the work area wizard. For custom objects, this combination of page types, configuration pages, a wizard-driven page creation process, and the ability to add links to the Navigator menu means that you can quickly and easily expose custom object model extensions to your users.

Before you begin to customize or create new pages for a Fusion CRM application, review the following aspects of the Application Composer Enterprise Page creation process:

- Using the Enterprise Pages Overview Page
- Understanding Page Types
- Defining Pages
- Creating a Work Area
- Object Security on Pages
- Using Oracle Composer to Customize Pages

Understanding Page Types

Every top-level CRM object can be displayed on a set of standard page types: a regional panes search page, an overview page, a creation page, and a details page.

- Regional Panes page
The Regional Panes page is the Enterprise page where you can specify the fields to use for a regional search.
- Overview page
The Overview page provides a list of records for an object and is the starting point in a CRM application for users to view and manage data. This page is where you can search for existing objects and can create new instances of an object. Users access an object's Overview page from the Navigator menu at runtime.

Note

Only top-level objects have an Overview page. If an object was created as a child to another object, then the child does not have an Overview page.

The Overview page includes two regions, each of which has its own configuration page:

- Summary table

The Summary Table includes a list of the object records. Depending on the security setup, users can create or edit an object, delete an object, or drill down into an existing record.

Tip

Optionally, define saved searches that users can select at runtime to constrain the list of records displayed in the Summary Table.

- Local search region

The local search region is displayed above the Summary Table. Users can enter search criteria to constrain the list of records that appear in the summary table.

- Creation page

The creation page is the Enterprise page where users can create new records for an object. Depending on the security setup, users access the creation page by clicking the New icon or by selecting the New menu item from the Actions menu on the summary table's toolbar.

- Details page

The details page is the left panel of the Enterprise page where users can view more details about an object. Depending on the security setup, users access the details page by clicking the Edit icon or by selecting the Edit menu item from the Actions menu on the summary table's toolbar. Users can also access the details page by clicking the object record name itself in the summary table.

The details page is the first page users see. It is basically a list of records and can include both a default summary and a detailed summary.

The default summary includes the primary object fields and is always displayed to users. The detailed summary includes additional fields for an object. You cannot add the same field to both the default and detailed summaries.

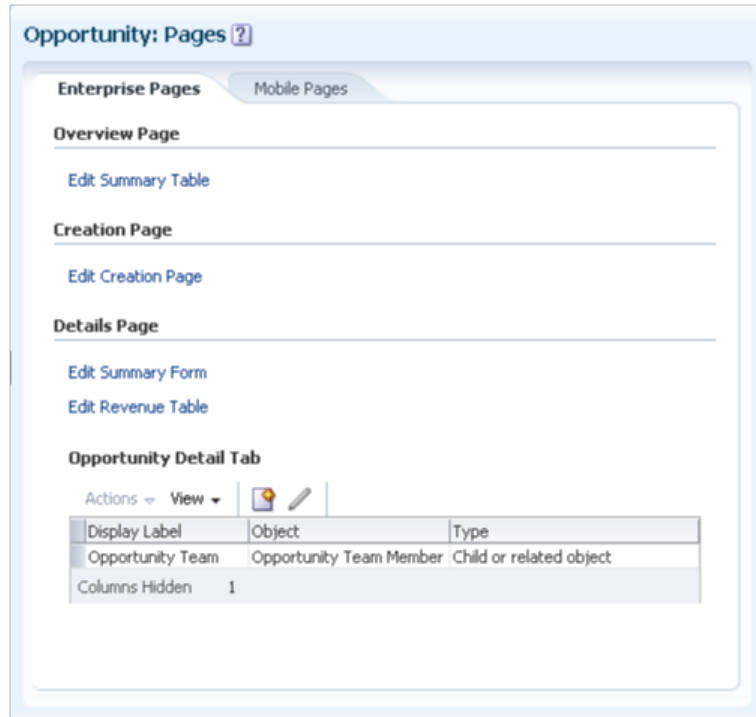
The details page can also display information related to the object record in subtabs. For example, the details page for an opportunity could include a subtab that lists customer contacts or previous orders.

Note

Some CRM objects, also known as common objects, do not have a standard work area. These include common components (note, interaction, task) and common objects (resource, organization contact, organization profile, address).

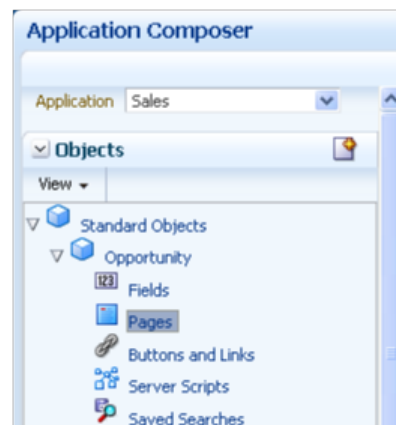
Using the Enterprise Pages Overview Page

The overview page in Application Composer provides an overview of the set of standard enterprise pages for an object.



To access the Pages overview page:

1. Select an application on the main Overview page.
2. Select a standard or custom object in the object tree.
3. Select the **Pages** node.



Note

Only top-level objects have pages that you can configure. A child object does not exist outside the context of the parent object, and does not have its own work area.

From the pages overview page, you can:

- View the pages where the object is already exposed to users, and further customize those pages by adding or removing fields.

- Create a new set of pages for an object, collectively known as a work area, if no set of pages has been created yet.
- Create one or more subtabs to display on the object's Details page.
- Create mobile application pages in Oracle Fusion Mobile Sales by using a wizard.

Similar to the work area creation process, the creation process for mobile pages uses a wizard where you can select which custom fields and related objects to add to mobile pages. Select the Mobile Pages tab to access the mobile pages wizard.

Defining Pages

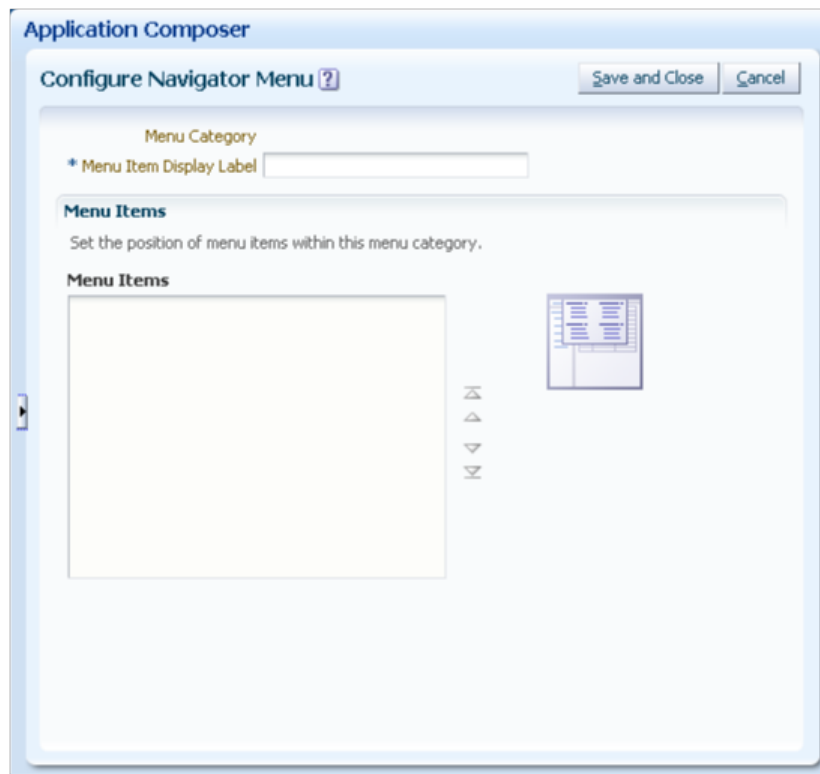
Use configuration pages to specify what fields are displayed on the standard pages for the object. After you create new objects and fields, navigate to the Pages Overview page. The Pages Overview page contains hyperlinks to the configuration pages for an object's existing work area. Use these configuration pages to customize the object's work area pages, for example add newly created fields to a creation page.

Note

If the Pages Overview page does not contain these configuration page hyperlinks, then the object does not yet have a work area, and you must create one if you want the object to be visible to users at runtime.

Use the configuration pages available from the Pages Overview page as follows:

- Navigator Menu
For custom objects, specify the object label that appears in the Navigator menu at runtime.



- Regional Panes page

Select the fields to appear in the regional search pane at runtime.

Configure Regional Search
Select the fields to appear in the regional search.
Enable ☒

Available Fields

- CreatedBy
- CreationDate
- CurrencyCode
- LastUpdateDate
- LastUpdatedBy

Selected Fields

- Sales Name

To display the entire regional search pane, check the Enable check box, otherwise it only shows the selected fields.

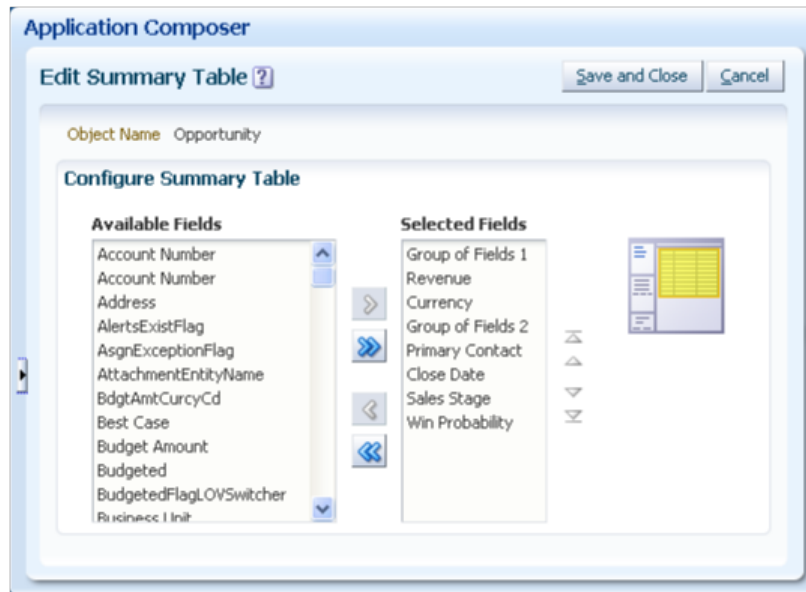
Selected Field	Default Operator	Required	At Least One is Required
Sales Name	Starts with ▼	<input type="checkbox"/>	<input type="checkbox"/>

For any field, you can select the following options:

- Required: The user must include this field.
- At Least One is Required: The user must include at least one of the selected fields.
- Default Operator: The user can define each field by using one of the listed operator options.
- Overview page

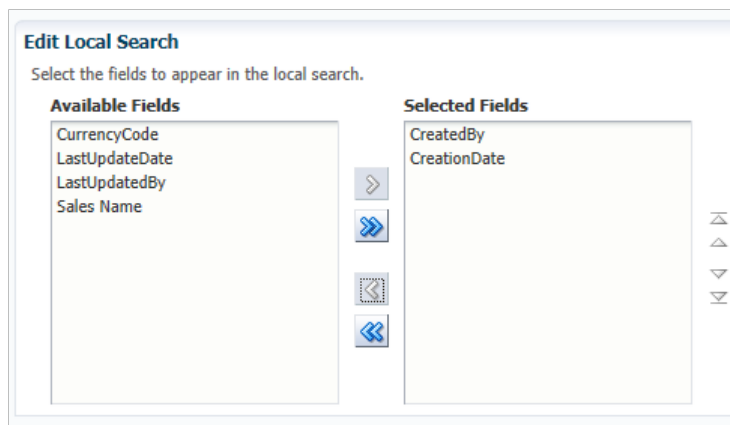
The overview page includes two regions:

- Summary table



1. Select the fields that you want to display as columns in the summary table.
2. Select the **Allow Access Grant** check box if you want users to be able to grant access to a record in the summary table to another user.
3. Add custom buttons to the summary table, if you previously created them.

- Edit Local Search page




1. Select the fields that will be used to search in the local search region.

The list of fields available for selection is displayed to you in a single column, although the local search is formatted as a region with two columns. The first field you select is displayed in the first column, the second field you select is displayed in the second column, the third field you select is displayed in the first column again, and so on.

Note

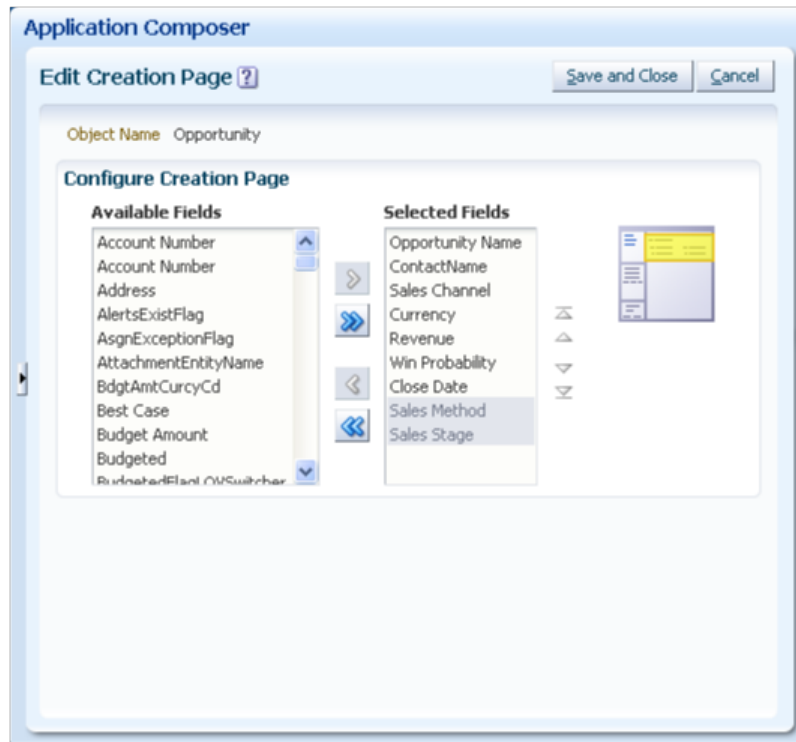
During field creation, consider indexing any fields that you plan to display as search criteria for your custom objects.



Selected Field	Default Operator	Required	At Least One is Required
CreatedBy	Starts with ▼	<input type="checkbox"/>	<input type="checkbox"/>
CreationDate	Equal to ▼	<input type="checkbox"/>	<input type="checkbox"/>
CurrencyCode	Starts with ▼	<input type="checkbox"/>	<input type="checkbox"/>
LastUpdateDate	Equal to ▼	<input type="checkbox"/>	<input type="checkbox"/>
LastUpdatedBy	Starts with ▼	<input type="checkbox"/>	<input type="checkbox"/>
Sales Name	Starts with ▼	<input type="checkbox"/>	<input type="checkbox"/>

2. For any field, you can select the following options:

- Required: The user must include this field.
 - At Least One is Required: The user must include at least one of the selected fields.
 - Default Operator: The user can define each field by using one of the listed operator options.
- Creation page



a. Select the fields that you want to display on the object's creation page.

The fields that you select must include the object's required fields.

The list of fields available for selection is displayed to you in a single column, although the creation page is formatted as a page with three columns. The first field you select is displayed in the first column, the second field you select is displayed in the second column, the third field you select is displayed in the third column, the fourth field you select is displayed in the first column again, and so on.

- Details page

The screenshot shows the 'Application Composer' window with the 'Edit Details Page Summary Form' dialog open. The dialog is titled 'Edit Details Page Summary Form' and has 'Save and Close' and 'Cancel' buttons. It shows the configuration for the 'Opportunity' object. There are two main sections: 'Configure Default Summary' and 'Configure Detailed Summary'. Each section has an 'Available Fields' list and a 'Selected Fields' list. In the 'Configure Default Summary' section, the 'Available Fields' list includes Account Number, Address, AlertsExistFlag, AsgnExceptionFlag, AttachmentEntityName, BdgAmtCurcyCd, Budget Amount, Budgeted, BudgetedFlagLOVSwitcher, and Business Unit. The 'Selected Fields' list includes Opportunity Name, Group of Fields 1, Owner, Revenue, Worst Case, Best Case, Currency, Status, Win/Loss Reason, Close Date, and Sales Method. In the 'Configure Detailed Summary' section, the 'Available Fields' list includes Account Number and Account Number. The 'Selected Fields' list includes CustomerAccountI and Opportunity Number.

- Select the fields that you want to display on the object's details page, including both the default summary and detailed summary regions.

Tip

Include the primary object fields in the default summary, since the detailed summary could be collapsed when users navigate to this page.

The list of fields available for selection is displayed to you in a single column, although the details page is formatted as a page with three columns. The first field you select is displayed in the first column, the second field you select is displayed in the second column, the third field you select is displayed in the third column, the fourth field you select is displayed in the first column again, and so on.

- Add custom buttons to the details page, if you previously created them.
- The Pages Overview page also lets you configure subtabs that display on the details page. Subtabs include information that is related to the object record. For example, the details page for an opportunity

could include a subtab that lists customer contacts or previous orders. Adding subtabs to a details page is discussed in a related topic.

If the object uses a tree to display related pages, rather than subtabs, then you can configure tree nodes that you add to the object's tree. Similar to subtabs, tree node data can be derived from another object, or from another source outside the current Oracle Fusion CRM application altogether. For more information on adding tree nodes to an object's tree, see Related Topics.

Note

As previously mentioned, some CRM objects (common components and common objects) do not have a standard work area. This means that the configuration pages available from their Pages Overview page are different than described previously. For example, the Trading Community address object has configuration pages for customizing the overview page, creation page, and address details form. The Trading Community organization profile has configuration pages for customizing only the details form and create form.

When you customize pages for common objects, the changes you make are reflected across the multiple applications where the object is used, provided that the applications also share the same metadata repository.

Creating a Work Area

When first created, top-level custom objects do not yet have pages in a runtime CRM application where those objects are exposed to users. For each custom object that you create, you must create the set of pages where the records that belong to the object will be exposed to users.

The Application Composer uses a wizard to walk you through the creation of these pages, also known as a work area. For more information on creating a work area, see Related Topics.

You do not create a work area for child objects.

Object Security on Pages

After you create custom objects and fields, you then expose them on Enterprise Pages for your users. By default, the object and its records are visible and can be edited only by a user who is assigned a duty role that is specified in the application. Grant additional access manually for either an object or role, using the Application Composer's security policy configuration pages.

The security options available to you for restricting access to custom objects, including child objects, are discussed in a related topic.

Using Oracle Composer to Customize Pages

Once you create a set of new pages, or edit preexisting pages delivered by a CRM application, you cannot use Oracle Composer to edit those pages again.

Note

The exception is the customer profile in Oracle Fusion Customer Center. You can create and add new fields to the Sales Account region on the customer profile using Oracle Composer.

Creating a Work Area: Explained

When first created, custom objects do not yet have pages in a runtime CRM application where those objects are exposed to users. After you create a top-level

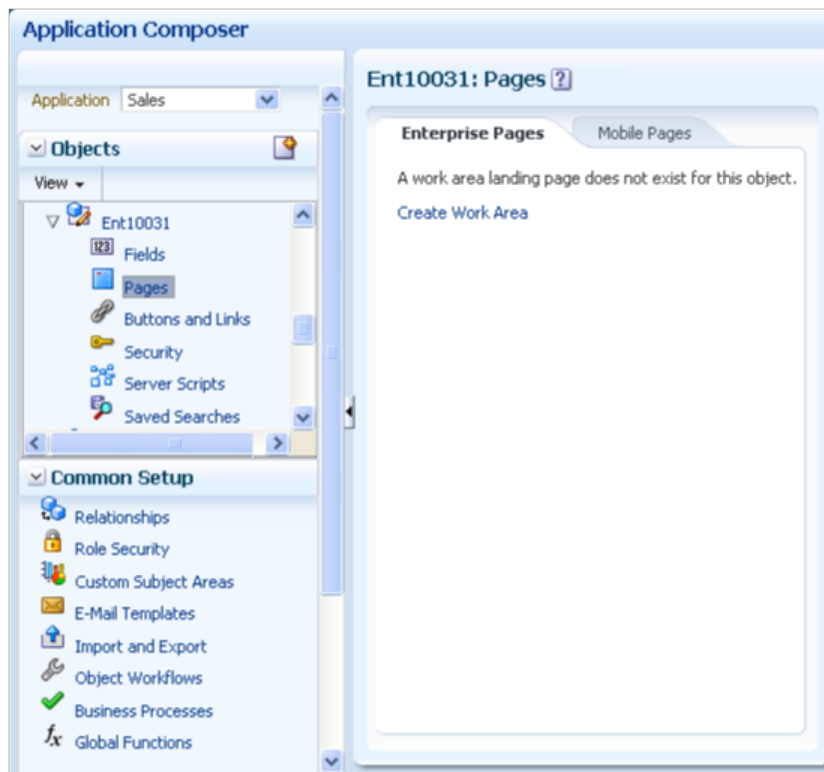
custom object, you must create a set of Enterprise pages, also known as a work area, for that object. Every CRM object can be displayed on a work area, which consists of an overview page, a creation page, and a details page. The Oracle Fusion CRM Application Composer employs a wizard approach to walk you through the creation of that object's work area. After you create the initial work area, you can always navigate to the object's Pages Overview page to continue to customize those Enterprise pages using work area configuration pages. You do not create a work area for child objects. To create and modify pages displayed on a mobile device, use the separate Mobile Pages wizard which is also available from the object's Pages Overview page.

Review these aspects of the work area creation process in the Application Composer before you create a new work area for a custom object:

- Using the work area wizard
- Configuring the Navigator menu
- Configuring the local search region
- Configuring the overview and creation pages
- Configuring the details page

Using the Work Area Wizard

Access the wizard on the Pages Overview page using the same navigation path that you use to configure pages in an existing work area. However, if a work area has not yet been created for an object, then hyperlinks to the work area configuration pages are not displayed. Instead, the Pages Overview page displays only a single hyperlink to launch the work area wizard.



To access the work area wizard:

1. Select an application on the main Overview page.
2. Select a standard or custom object in the object tree.
3. Select the **Pages** node.

Note

Only top-level objects have pages that you can configure. A child object does not exist outside the context of the parent object, and does not have its own work area.

-
4. Select the hyperlink to launch the work area wizard.

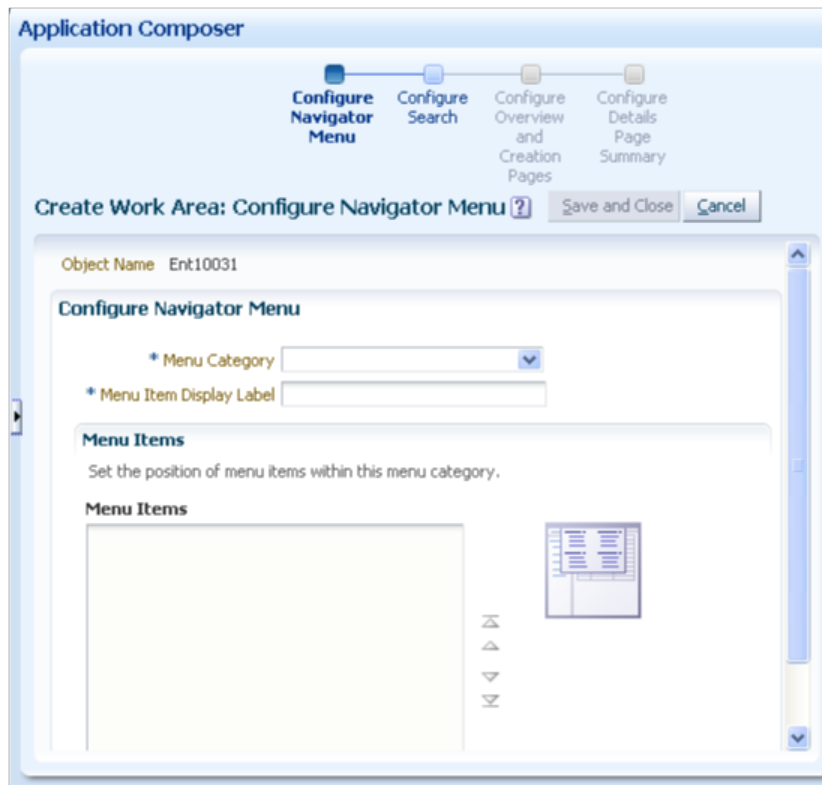
Note

Use the work area wizard to create a work area.

Use the work area configuration pages to customize existing work area pages.

Configuring the Navigator Menu

As part of the work area creation for a custom object, you must specify the object label that appears in the Navigator menu at runtime. The label you specify is what users will select to navigate to this work area.



On this page, you can also do the following:

- Select a menu category under which the object label appears.

- Adjust the position of Navigator menu items within the selected menu category.

For example, move your newly created object label to appear at the top of the list.

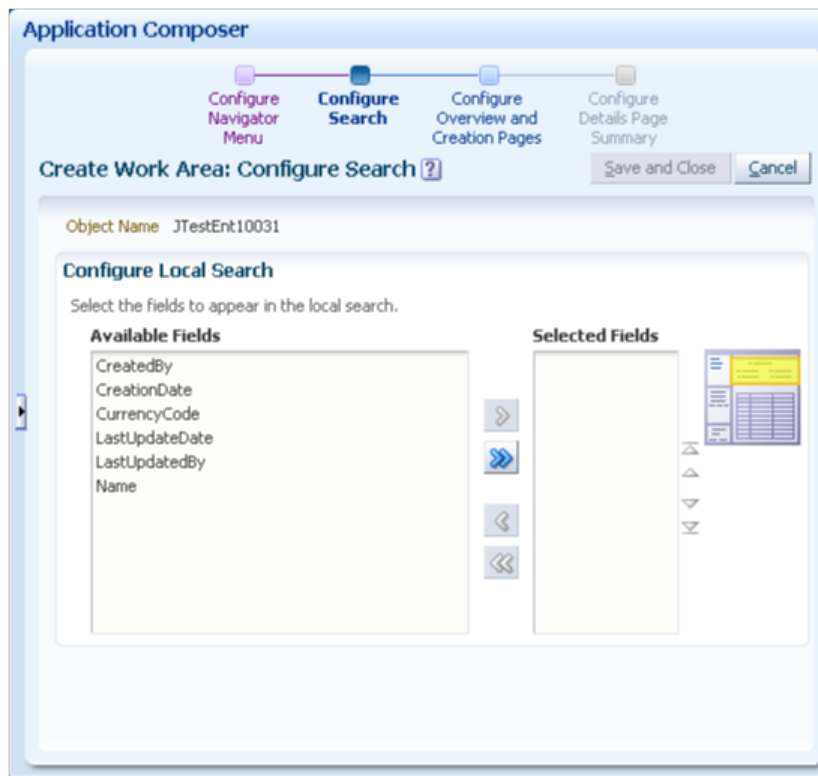
After creating the work area for a custom object, the work area label automatically appears in the Navigator menu without your having to reauthenticate.

Note

Changing the object label on the Navigator menu is available in the Application Composer only for custom objects. To change the menu label for a standard object, refer to Fusion Setup Manager documentation.

Configuring the Local Search Region

Select the fields that you want to display as search criteria in the local search region. The local search region appears above the summary table on an object's overview page. Adding fields to this region is optional.



1. Select the fields that you want to display as search criteria fields in the local search region.

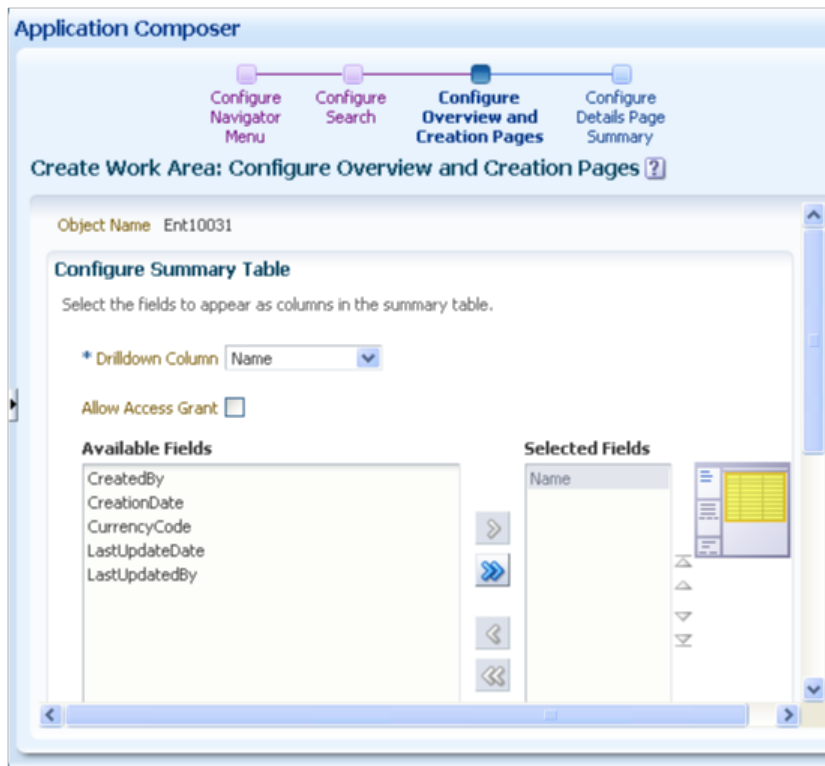
The list of fields available for selection is displayed to you in a single column, although the local search region is formatted as a region with two columns. The first field you select is displayed in the first column, the second field you select is displayed in the second column, the third field you select is displayed in the first column again, and so on.

Note

During field creation, consider indexing any fields that you plan to display as search criteria for your custom objects.

Configuring the Overview and Creation Pages

Select the fields that you want to display in the work area's overview page, and in the object's creation page.



1. Select the fields that you want to display as columns in the summary table, on the object's overview page.
2. Select the drilldown column for the summary table.

The drilldown column is the column in the summary table that users can click to drill down to an object record's details page. You cannot change a summary table's drilldown column after the work area is created.

3. Select the **Allow Access Grant** check box if you want users to be able to grant access to a record in the summary table to another user, at runtime.
4. Add custom buttons to the summary table, if you previously created them.
5. Select the fields that you want to display on the object's creation page.

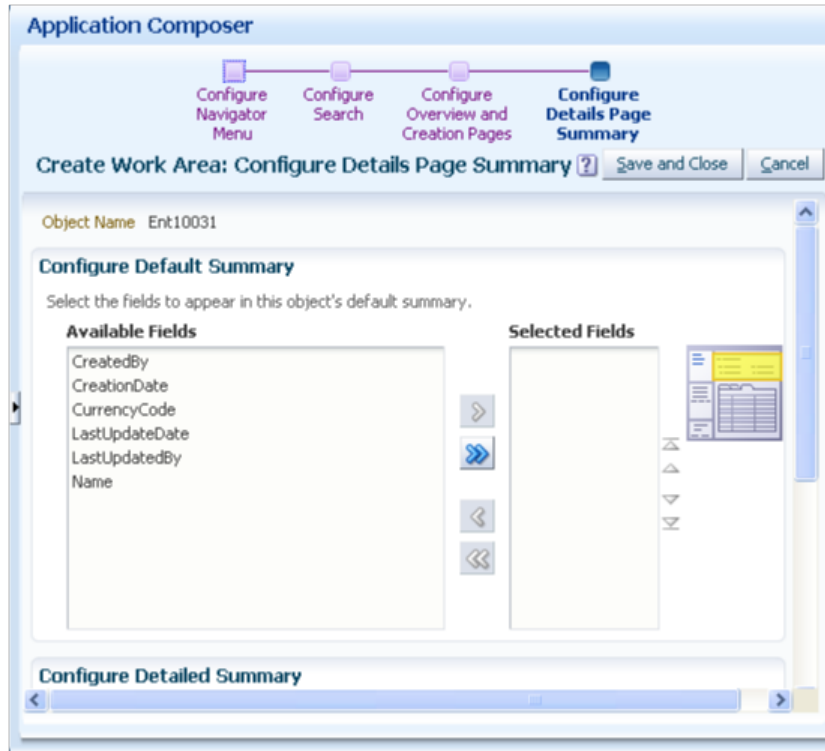
The fields that you select should include the object's required fields.

The list of fields available for selection is displayed to you in a single column, although the creation page is formatted as a page with three columns. The first field you select is displayed in the first column, the second field you select is displayed in the second column, the third field

you select is displayed in the third column, the fourth field you select is displayed in the first column again, and so on.

Configuring the Details Page

Select the fields that you want to display on the object's details page.



Note

A details page can have subtabs, which include information that is related to the object record. For example, the details page for an opportunity could include a subtab that lists customer contacts or previous orders. To add subtabs to a details page, create the work area first, then navigate back to the Pages Overview page. Adding subtabs to a details page is discussed in a related topic.

-
1. Select the fields that you want to display on the object's details page, including both the default summary and detailed summary regions.

Tip

Include the primary object fields in the default summary, since the detailed summary could be collapsed when users navigate to this page.

The list of fields available for selection is displayed to you in a single column, although the details page is formatted as a page with three columns. The first field you select is displayed in the first column, the second field you select is displayed in the second column, the third field you select is displayed in the third column, the fourth field you select is displayed in the first column again, and so on.

2. Add custom buttons to the details page, if you previously created them.

3. Select the **Allow Attachments** check box to enable the attachments feature on the runtime details page, in the collapsible detailed summary.

Subtabs: Explained

Every top-level CRM object has a details page as part of its work area. When configuring the details page, you can optionally display details that are related to the current object but derived from another object, or from another source outside the current Oracle Fusion CRM application altogether. You do this by adding subtabs to the details page, and specifying the source of subtab data. Add subtabs to a standard or custom object's details page from that object's Pages Overview page in the Oracle Fusion CRM Application Composer.

Review these aspects of the subtab creation process in the Application Composer before you begin to add subtabs to an object's details page:

- Using the Details page
- Adding subtabs
- Subtab types:
 - Child or related object subtabs
 - Context link subtabs
 - Common component subtabs
 - Web content subtabs

Note

Subtabs and tree nodes are two master/detail UI patterns which Oracle Fusion CRM applications support.

For custom objects, only subtabs are supported.

For standard objects that are already using tree nodes, such as the Sales Account Profile and Partner objects, additional details adopt the same tree node pattern. In other words, if a standard object uses a tree to display its related pages, then you would expose child or related objects, for example, as tree nodes instead of subtabs on a details page. Adding tree nodes is discussed in a related topic.

Using the Details Page

The details page is the Enterprise page where users can view more details about an object. Depending on the security setup, users access the details page by clicking the Edit icon or by selecting the Edit menu item from the Actions menu on the summary table's toolbar. Users can also access the details page by clicking the object record name itself in the summary table.

The details page can include both a default summary and a detailed summary. The default summary includes the primary object fields and is always displayed to users. The detailed summary includes additional fields for an object. You cannot add the same field to both the default and detailed summaries.

The details page can also include information that is related to the object record, and displayed in subtabs. For example, the details page for an opportunity could include a subtab that lists customer contacts or previous orders.

Adding Subtabs

Add a subtab to an object's details page from that object's Pages Overview page. The details page must exist already; you cannot add subtabs when first creating a work area.

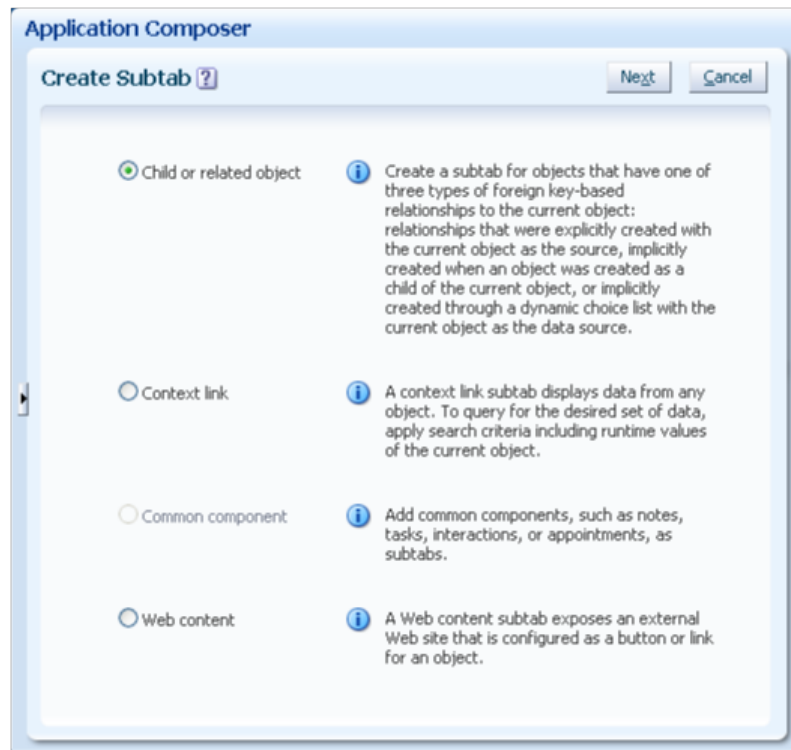
To add a subtab to an existing details page:

1. Select an application on the main Overview page.
2. Select a standard or custom object in the object tree.
3. Select the **Pages** node.

Note

Only top-level objects have pages that you can configure. A child object does not have its own work area.

4. On the Pages Overview page, click the **Create Subtab** icon in the Details Page region to create one or more subtabs to display on the object's details page.
5. Select the type of subtab you want to add.



Child or Related Object Subtabs

A relationship is a foreign key association between two objects. Using the Application Composer, you can create a one-to-many relationship between two objects within the same application. Once relationships are created, you can expose the "many" objects on a subtab that is displayed on the "one" object's details page. For example, an account can have multiple service requests

associated to it. To expose a list of service requests associated with a specific account as a subtab on the account's details page, you must first create a one-to-many relationship between the account and service request objects. In this example, the account is the source object and the service request is the target object. This relationship adds the account identifier to the service request object's table.

The Application Composer lets you add a subtab to an object's details page for either a child object or for three types of related objects. These objects exist in four types of one-to-many relationships:

- Parent child relationship

Parent child relationships are implicitly created when a custom object is created as a child of a top-level object.

For example, to enable the creation of a subaccount subtab on an account's details page, you would create the subaccount object as a child of the account object. This relationship adds the account identifier to the subaccount object's table.

- Choice list relationship

Choice list relationships are implicitly created between two objects when you create a dynamic choice list field.

For example, to enable the creation of a department subtab on an employee's details page, you would create a dynamic choice list, HR Representative, for the department object where the choice list's related object is employee. The Application Composer automatically creates the underlying relationship for you, where the employee is the source object and the department is the target object. This relationship adds the employee identifier to the department object's table, thus enabling the creation of a department subtab on an employee's details page. The subtab displays all departments that an HR representative can manage, since each HR representative can be in charge of multiple departments of a company.

- Reference relationship

Reference relationships are explicitly created between two top-level objects using the Create Relationships page.

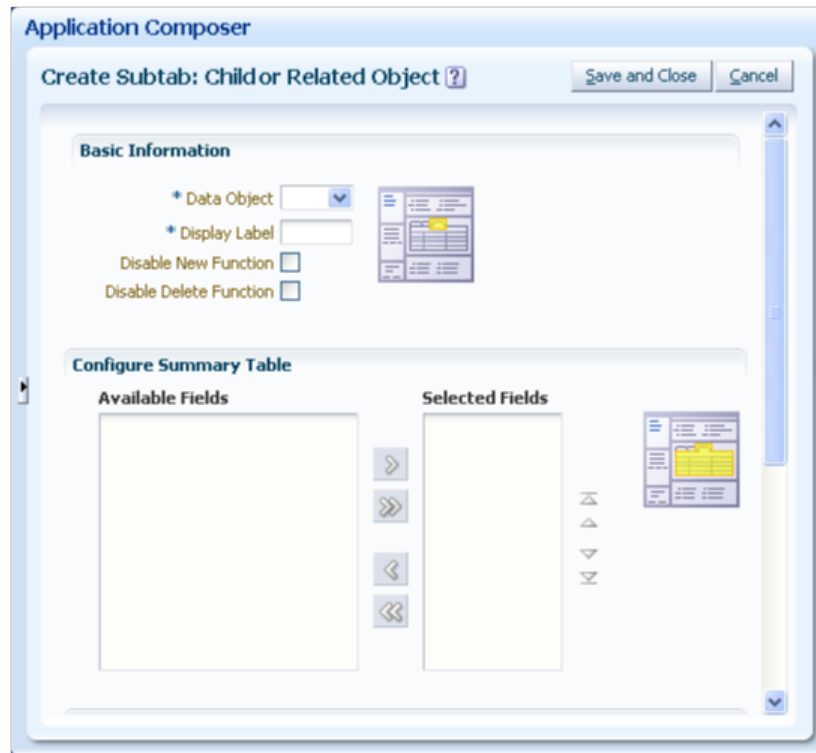
Using our previous example, perhaps you don't need to display an HR Representative choice list on a department Enterprise page, but you still want to add a department subtab to an employee's details page. In this case, manually create a reference relationship between the employee and department objects where the employee is the source object and the department is the target object. This enables the creation of the department subtab. Such a reference relationship, however, does not automatically create a corresponding HR Representative choice list for use on the department Enterprise page. In fact, once you manually create a relationship, you cannot reuse the relationship to create a choice list. This means that you should carefully consider the need for a choice list before you create a reference relationship.

- Standard relationship

Standard relationships are relationships that are already created between two standard objects by the Oracle Fusion CRM application.

To add a child or related object subtab to an existing details page:

1. On the Pages Overview page, click the **Create Subtab** icon.
2. Select **Child or Related Object**.



3. On the Child or Related Object subtab configuration page:
 - a. Select the related object from the list of all related objects that is to be exposed on the subtab, and choose the subtab display label.
 - b. Optionally hide the New and Delete buttons that appear on the subtab at runtime.

For child object subtabs, you can also optionally hide the Edit button.

- c. For child object subtabs only, specify if you want to display the **Manage Permission** button on the subtab at runtime.

At runtime, users can select an object record and click that button to specify the level of access another user should have to the selected record.

- d. Select which fields and links you want to display on the subtab summary table at runtime.

You can configure fields and links for the main summary table which lists the child object records or related object records.

- e. Select which buttons you want to display on the subtab at runtime.

Note

This region appears only if you previously created buttons for this object.

- f. Select which fields you want to display on the subtab detail form at runtime.

You can configure fields for the detail form that appears under the summary table. If the subtab's object is a child object, then users can enter child object data into this detail form at runtime. Always include required fields in this section.

If the subtab's object is a related object, then users can associate an existing record of the subtab object to the master object of the page. However, to create new related object records, users must do so in the object's own creation page.

Context Link Subtabs

A context link subtab displays a filtered list of records from any top-level object, where the filter is often based on the runtime values from the current object. The object does not have to be related to the current object. Context link subtabs are read only.

To add a context link subtab to an existing details page:

1. On the Pages Overview page, click the **Create Subtab** icon.
2. Select **Context Link**.

The screenshot shows the 'Create Subtab: Context Link' configuration window in Oracle Application Composer. The window has a title bar 'Application Composer' and a subtitle 'Create Subtab: Context Link'. It includes 'Save and Close' and 'Cancel' buttons. The configuration is divided into three main sections: 'Basic Information', 'Search Criteria', and 'Configure Summary Table'. The 'Basic Information' section contains fields for 'Data Object' (a dropdown menu) and 'Display Label' (a text field). The 'Search Criteria' section has a 'Match' section with radio buttons for 'All' (selected) and 'Any'. Below this is a table with four rows (A, B, C, D) and four columns: 'Condition', 'Field Name', 'Operator', and 'Value'. Each row has dropdown menus for 'Field Name' and 'Operator', and a 'Value' field with a 'Literal value' dropdown. The 'Configure Summary Table' section has two panes: 'Available Fields' and 'Selected Fields', with a right-pointing arrow button between them. A small thumbnail of the subtab is visible in the bottom right corner.

3. On the Context Link subtab configuration page:
 - a. Select the object that is to be exposed on the subtab, and choose the subtab display label.

- b. Optionally constrain the list of records displayed at runtime using a set of search criteria for the selected object, whose runtime values must match the current object record's runtime values.

Tip

Values can be literal values, or derived from the runtime values in the current object record, or from the runtime values in the current object's parent record.

If your search criteria includes a fixed choice list field, then you must specify the fixed choice list's runtime value using the lookup code, not the lookup meaning.

-
- c. Select which fields you want to display on the subtab's read-only summary table at runtime.

You can configure fields for the main summary table which lists the child object records or related object records.

- d. Select which fields you want to display on the subtab's read-only detail form at runtime.

You can configure fields for the detail form that appears under the summary table.

Common Component Subtabs

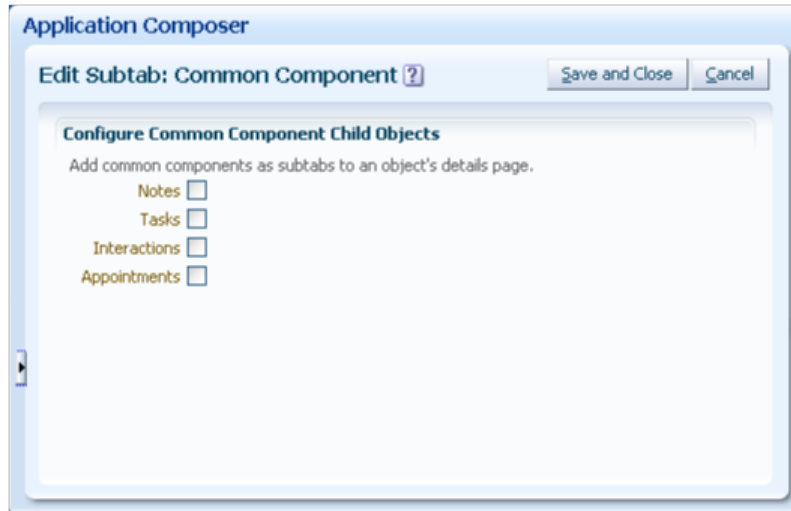
A common component subtab adds a Notes, Tasks, Interactions or Appointments subtab to show a list of the selected components related to a custom, top-level object. Each component has a standard user interface (UI) that is shared across all Oracle Fusion CRM applications. To customize such a UI for all common components (other than Appointments), select the appropriate object under the **Common** application, then select the Pages node on the object's navigation tree to access the work area configuration pages.

At runtime, users can access these subtabs and create a common component record that is tied to the object record. For example, a user can record a customer interaction on an service request record.

- Notes
- Tasks
- Interactions
- Appointments

To add a common component subtab to an existing details page:

1. On the Pages Overview page, click the **Create Subtab** icon.
2. Select **Common Component**.



3. On the Common Component subtab configuration page, select the type of common component you want to add to the details page as a subtab.

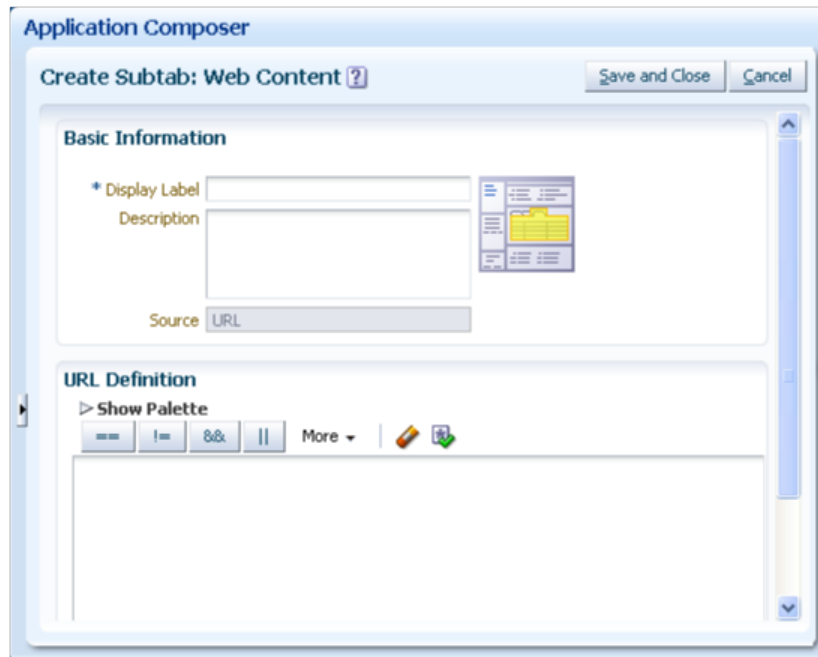
Web Content Subtabs

A Web content subtab exposes an external Web site right on an object's details page. The Web content is a result of the expression that you define which builds the intended URL.

For example, on the Contact details page, perhaps you want to add a Google map that shows the location of the contact. The Google Maps API expects the URL to be formatted in a certain manner. In this example, write an expression using the fields: Contact Address, Contact City and Contact State. Then, pass the URL to the Google Maps API.

To add a Web content subtab to an existing details page:

1. On the Pages Overview page, click the **Create Subtab** icon to create one or more subtabs to display on the object's details page.
2. Select **Web Content**.



3. On the Web Content subtab configuration page, enter the display label for the subtab, and then define the URL to retrieve the subtab's Web content. Optionally use the expression editor to build the URL expression that you need.

The expression you build should include the following:

- Use the HTTP protocol.
- Optionally include field values from the current object as parameters, or user variables.
- Enclose static strings in double quotation marks.
For example, "http://www.abc.com/".

For example:

```
def myURL1 = adf.util.GlobalEncodeField(ContactAddress_c)
def myURL2 = adf.util.GlobalEncodeField(ContactCity_c)
def myURL3 = adf.util.GlobalEncodeField(ContactState_c)
def myfinalURL = "http://maps.google.com/maps?hl=en&q=" + myURL1 + "+" +
  myURL2 + "+" + myURL3
return(myfinalURL)
```

Tree Nodes: Explained

Some CRM standard objects, such as the Sales Account Profile and Partner objects, use a tree to display its related pages. When configuring an object's work area, you can optionally display details that are related to the current object by adding tree nodes to the object's tree, and specifying the source of tree node data. Tree node data can be derived from another object, or from another source outside the current Oracle Fusion CRM application altogether. Add a tree node to a standard object's tree from that object's Pages Overview page in the Oracle Fusion CRM Application Composer.

Review these aspects of the tree node creation process in the Application Composer before you begin to add tree nodes to an object's tree:

- Adding tree nodes
- Tree node types:

- Child or related object tree nodes
- Context link tree nodes
- Web content tree nodes

Note

Subtabs and tree nodes are two master/detail UI patterns which Oracle Fusion CRM applications support.

For custom objects, only subtabs are supported.

For standard objects that are already using tree nodes, such as the Sales Account Profile and Partner objects, additional details adopt the same tree node pattern. In other words, if a standard object uses a tree to display its related pages, then you would expose child or related objects, for example, as tree nodes instead of subtabs on a details page. Adding subtabs is discussed in a related topic.

Adding Tree Nodes

Add a tree node to an object's tree from that object's Pages Overview page.

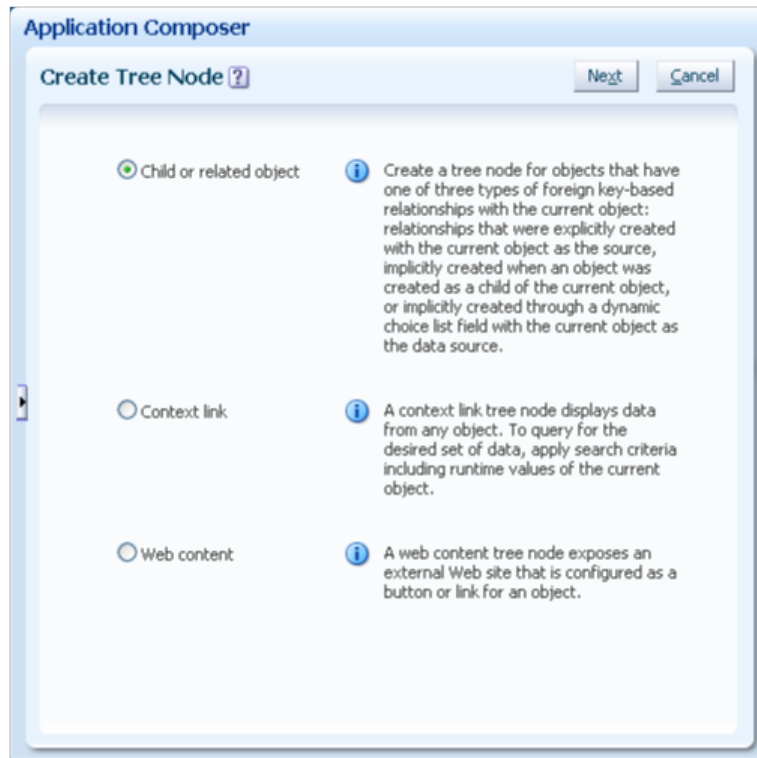
To add a tree node to an object's tree:

1. Select an application on the main Overview page.
2. Select a standard object, either the Sales Account Profile or Partner object, in the object tree.
3. Select the **Pages** node.

Note

Only the top-level objects, Sales Account Profile and Partner, let you add tree nodes.

-
4. On the Pages Overview page, click the **Create Tree Node** icon to create one or more tree nodes to display on the object's tree.
 5. Select the type of tree node you want to add.



Child or Related Object Tree Nodes

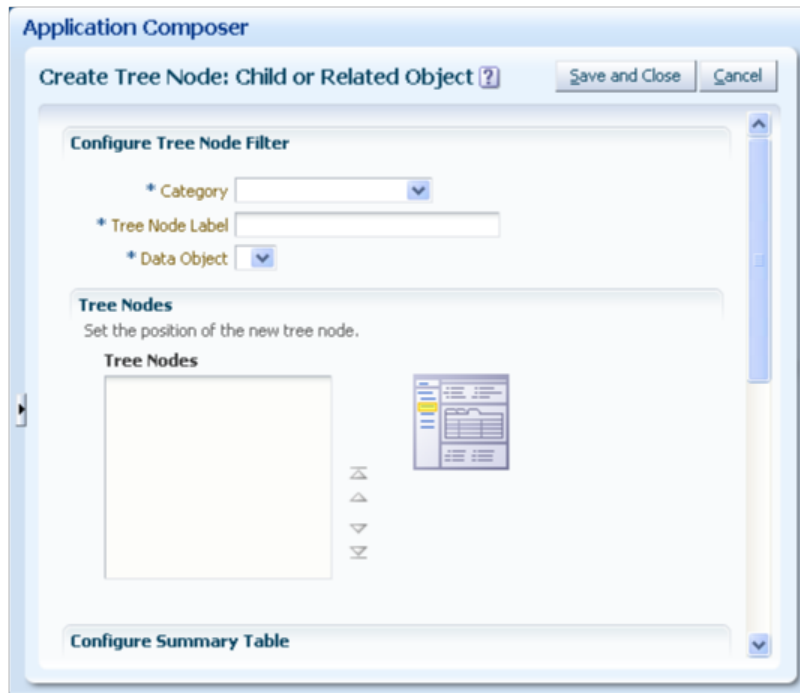
A relationship is a foreign key association between two objects. Using the Application Composer, you can create a one-to-many relationship between two objects within the same application. Once relationships are created, you can expose the "many" objects on a tree node that is displayed on the "one" object's tree. For example, a partner can have multiple contacts associated to it. To expose a list of contacts associated with a specific partner as a tree node on the partner's tree, you must first create a one-to-many relationship between the partner and contact objects. In this example, the partner is the source object and the contact is the target object. This relationship adds the partner identifier to the contact object's table.

The Application Composer lets you add a tree node to an object's tree for either a child object or for three types of related objects. These objects exist in four types of one-to-many relationships, which are described in detail in the related topic about object relationships:

- Parent child relationship
- Choice list relationship
- Reference relationship
- Standard relationship

To add a child or related object tree node to an existing tree:

1. On the Pages Overview page, click the **Create Tree Node** icon.
2. Select **Child or Related Object**.



3. On the Child or Related Object tree node configuration page:
 - a. Select the tree node category and enter the tree node label.
 - b. Select the related object from the list of all related objects that is to be exposed on the tree node page.
 - c. Set the position of the new tree node.
 - d. Optionally hide the New and Delete buttons that appear on the tree node page at runtime.

For child object tree node pages, you can also optionally hide the Edit button.

- e. For child object tree node pages only, specify if you want to display the **Manage Permission** button on the tree node page's summary table at runtime.

At runtime, users can select an object record and click that button to specify the level of access another user should have to the selected record.

- f. Select which fields and links you want to display on the tree node page's summary table at runtime.

You can configure fields and links for the main summary table which lists the child object records or related object records.

- g. Select which buttons you want to display on the tree node page at runtime.

Note

This region appears only if you previously created buttons for this object.
You cannot add buttons to a tree node page for the Sales Account Profile object.

- h. Select which fields you want to display on the tree node page's detail form at runtime.

You can configure fields for the detail form that appears under the summary table. If the tree node's object is a child object, then users can enter child object data into this detail form at runtime. Always include required fields in this section.

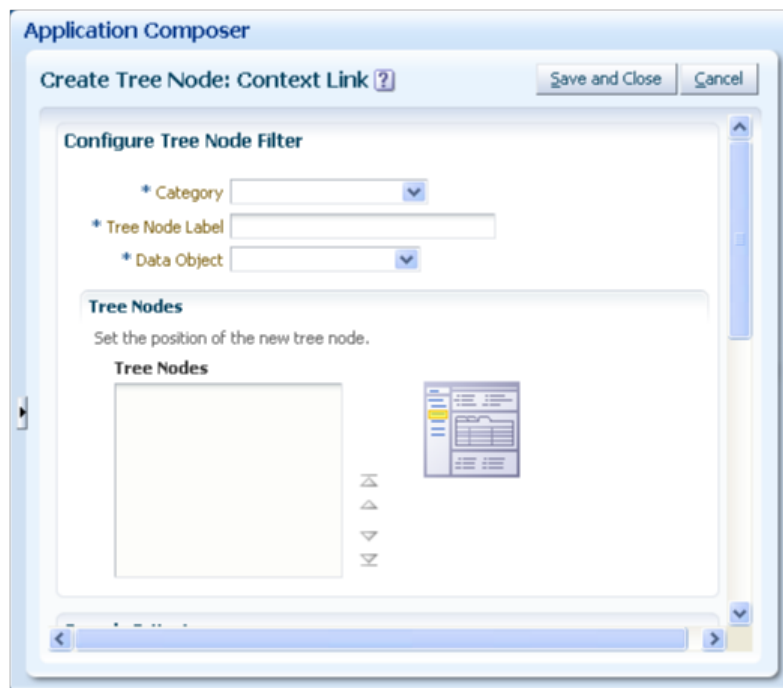
If the tree node's object is a related object, then users can associate an existing record of the tree node object to the master object of the page. However, to create new related object records, users must do so in the object's own creation page.

Context Link Tree Nodes

A context link tree node page displays a filtered list of records from any top-level object, where the filter is often based on the runtime values from the current object. The object does not have to be related to the current object. Context link tree node pages are read only.

To add a context link tree node to an object's tree:

1. On the Pages Overview page, click the **Create Tree Node** icon.
2. Select **Context Link**.



3. On the Context Link tree node configuration page:
 - a. Select the tree node category and enter the tree node label.
 - b. Enter the name of the tree node filter.

- c. Select the object that is to be exposed on the tree node page.
- d. Set the position of the new tree node.
- e. Optionally constrain the list of records displayed at runtime using a set of search criteria for the selected object, whose runtime values must match the current object record's runtime values.

Tip

Values can be literal values, or derived from the runtime values in the current object record, or from the runtime values in the current object's parent record.

If your search criteria includes a fixed choice list field, then you must specify the fixed choice list's runtime value using the lookup code, not the lookup meaning.

- f. Select which fields you want to display on the tree node page's read-only summary table at runtime.

You can configure fields for the main summary table which lists the child object records or related object records.

- g. Select which fields you want to display on the tree node page's read-only detail form at runtime.

You can configure fields for the detail form that appears under the summary table.

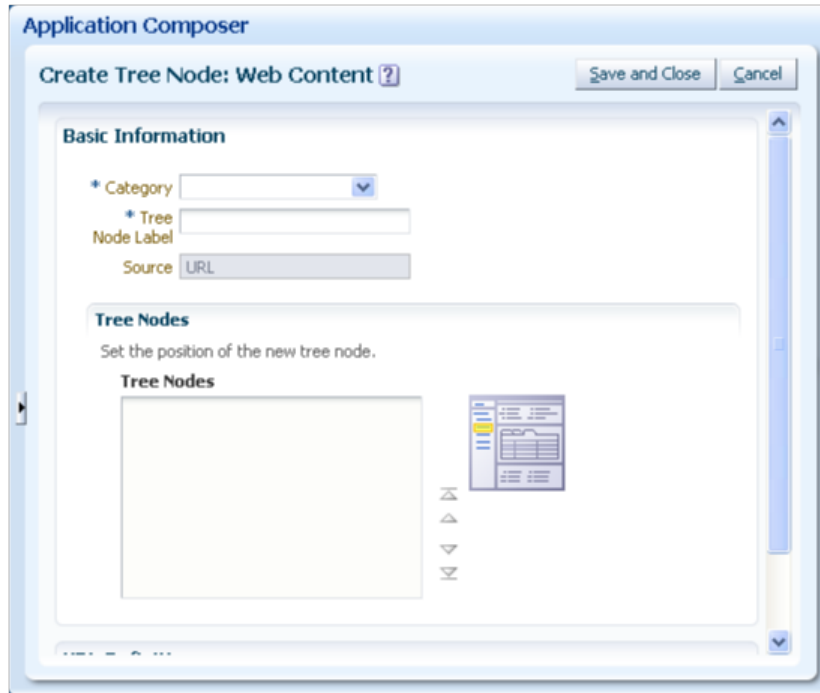
Web Content Tree Nodes

A Web content tree node page exposes an external Web site on an Enterprise page. The Web content is a result of the expression that you define which builds the intended URL.

For example, on the Contact tree node page, perhaps you want to add a Google map that shows the location of the contact. The Google Maps API expects the URL to be formatted in a certain manner. In this example, write an expression using the fields: Contact Address, Contact City and Contact State. Then, pass the URL to the Google Maps API.

To add a Web content tree node to object's tree:

1. On the Pages Overview page, click the **Create Tree Node** icon to create one or more tree nodes to display on the object's tree.
2. Select **Web Content**.



3. On the Web Content tree node configuration page:
 - a. Select the tree node category and enter the tree node label.
 - b. Set the position of the new tree node.
 - c. Define the URL to retrieve the tree node page's Web content.

Optionally use the expression editor to build the URL expression that you need.

The expression you build should include the following:

- Use the HTTP protocol.
- Optionally include field values from the current object as parameters, or user variables.
- Enclose static strings in double quotation marks.

For example, "http://www.abc.com/".

For example:

```
def myURL1 = adf.util.GlobalEncodeField(ContactAddress_c)
def myURL2 = adf.util.GlobalEncodeField(ContactCity_c)
def myURL3 = adf.util.GlobalEncodeField(ContactState_c)
def myfinalURL = "http://maps.google.com/maps?hl=en&q=" + myURL1 + "+" +
  myURL2 + "+" + myURL3
return(myfinalURL)
```

Actions and Links: Explained

In Oracle Fusion CRM applications you can add actions, such as scripts, and buttons to detail pages, list pages, and so on. You can also create special

fields, rendered as links, that are displayed with other fields throughout the application.

Actions

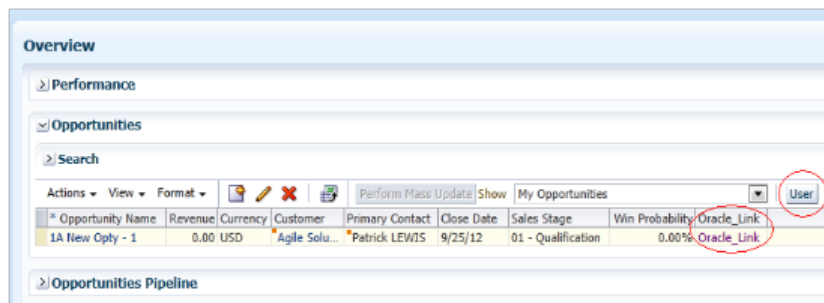
An action can be based on a script (a Groovy method that is defined on the object) or a URL. When configuring the work area for a standard or custom object, you can add custom links or actions to a page-level or task-level Actions menu or as a toolbar button. You can also manage the Actions menu by hiding or showing menu items, rearranging the action groupings or display sequence, and managing the toolbar by hiding or showing icons and buttons. You can also configure the Actions menu and buttons in the Create and Edit subtabs and tree nodes. A button can perform an action or navigate the user to another page in the runtime application, or to another web site. For example, you might want to provide a static link from an overview page to a corporate web site. Or, you might want to include a button on a summary table, which users can click at runtime to create a new type of record from a selected row, such as escalating an existing "trouble ticket" to a more severe "case" that can be managed separately. After you create an action, it can be exposed as a button or an option on the Action menu. After you create a link, it can be selected as a field.

Adding Actions or Links

You add actions or links in two steps:

1. Define an action or link for an object.
2. Use the Oracle Fusion CRM Application Composer's work area configuration pages to add that action or link to an Overview page or Details page.

The following figure shows a button and a link added to the Sales Opportunities Overview page.

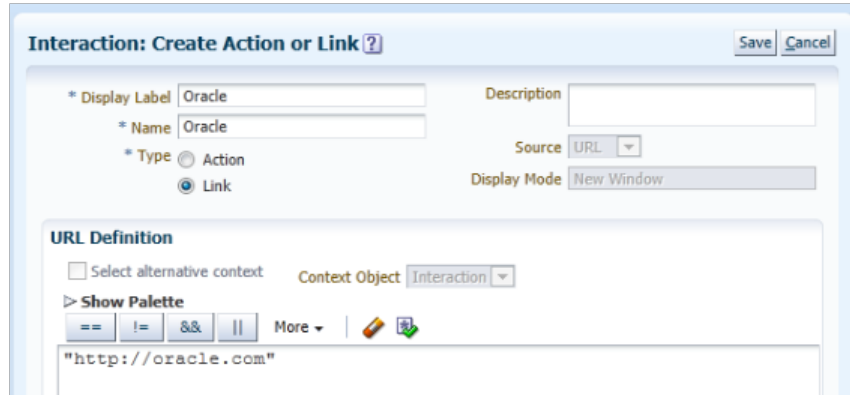


To define an action or link for an object:

1. Select an application on the main Overview page.
2. Select a standard or custom object in the object tree.
3. Select the **Actions and Links** node.

To create a new script or URL:

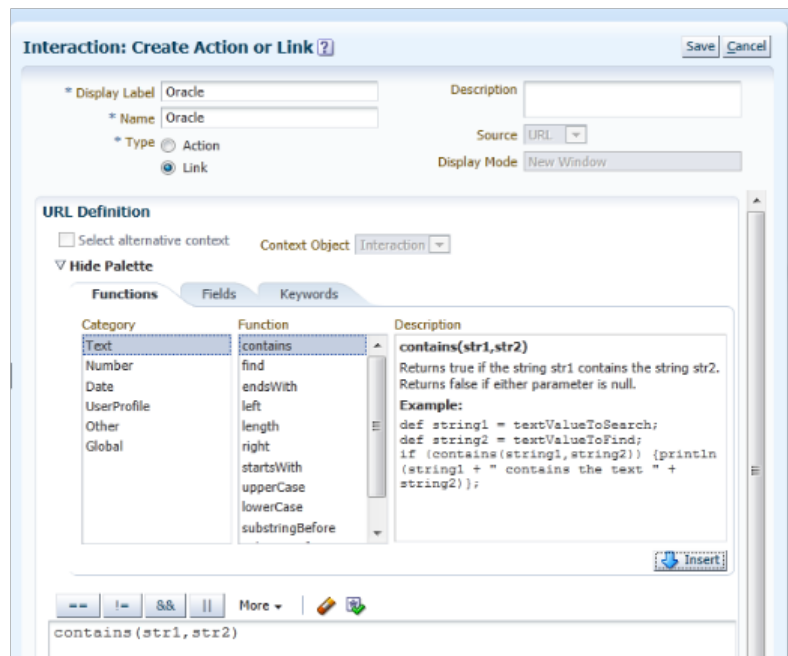
1. In the Create Action or Link page, enter a descriptive name in the Display Label field.
2. For Type, select **Action** and, for Source, select **Script** or **URL**.
3. In the Script region click the **New** icon.



If the source is a URL, you can enter a static URL, enclosed in double quotation marks. Or, you can define the URL by using the expression editor, which provides access to this object's fields to assist you in constructing the URL. If this object has a parent or relationship with a source object, then optionally change the context to access another object's fields for URL definition.

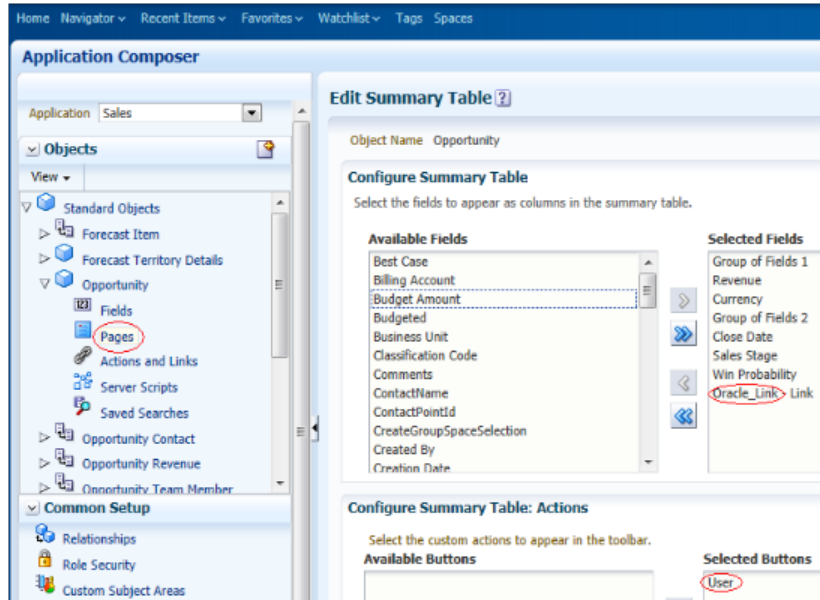
Any new functions that you create will be added to the **Method Name** choice list. If functions were already created for the object, then you can select one of them from the Method Name choice list. Object functions that are created elsewhere through other flows, such as server scripts, can also be used here.

To switch the context to the object's parent or related source object, for access to the object's fields for the URL definition, check the **Select alternative content** check box.

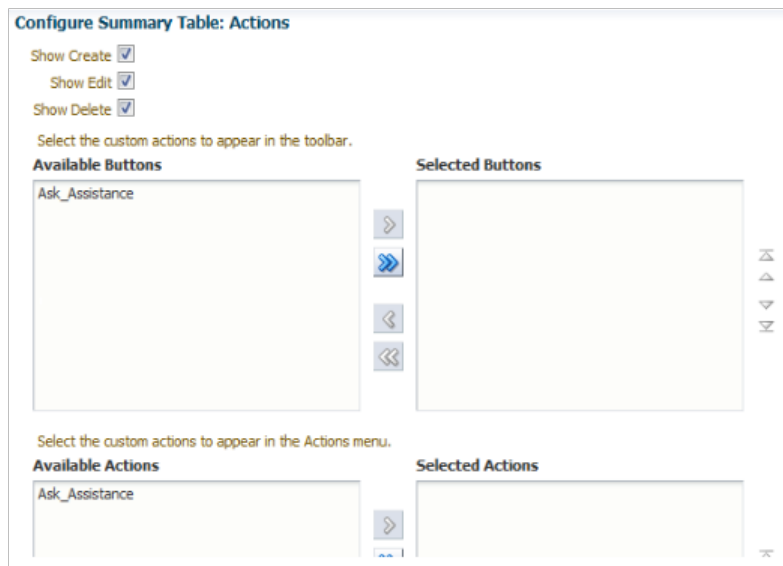


If the source is a script, you can either select a predefined object function from the **Method Name** choice list, or create a new object function using the expression editor.

After you save actions or links, you can expose them on UI pages by configuring the Applications Composer options available in the Edit Summary Table page in the Pages node of an object.



For example, the Edit Summary Table could have links in the fields selector with standard or custom fields because at runtime the UI displays the URL link as if it is a field in a table. Actions can be configured in potentially two places in the UI, on the toolbar as a button and in the Actions menu for a table. Depending on how you configure the actions and links, in the Edit Summary Table you could see both the available and selected buttons and actions, or one, or none.

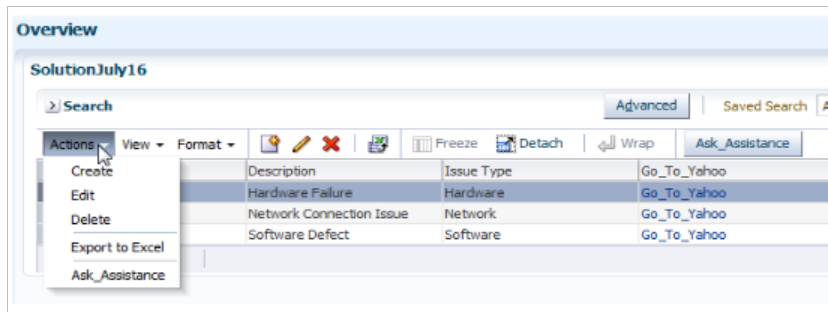


The figure above shows the Configure Summary Table: Actions with options checked for the Show Create, Show Edit, and Show Delete options on the Action menu. It also shows a custom button and a custom action.

After you define an action, you can then expose it as a button or an Action menu option in a variety of locations:

- Summary table on the overview page
- Default summary on the details page
- Summary table on a details page's subtab

- Summary table on a tree node page for a child object
- Revenue table on the details page for the opportunity object



The figure above shows the exposed Create, Edit, and Delete options and a custom Ask_Assistance option on the Actions menu. It also shows the custom toolbar button Ask_Assistance, and a custom table column.

After you define a link for an object, you can add that link to a variety of locations in that object's work area. You can add a link wherever you can add a field. Possible locations include, but are not limited to:

- As a column in the summary table on the overview page
- Default summary on the details page
- As a column in the summary table on a details page's subtab
- In the detail form under the summary table on a details page's subtab
- As a column in the summary table on a tree node page for a child object
- As a column in the revenue table on the details page for the opportunity object

Saved Searches for CRM Objects: Explained

A saved search is a predefined set of search criteria that you define which users can apply at runtime to a standard or custom object's overview page. The overview page provides a list of records for an object in a summary table, which is the starting point in a CRM application for users to view and manage data. If you think your users will be most interested in a particular data set (for example, the top 10 opportunities), then define a saved search for an object. At runtime, users can select a saved search from the **Saved Search** choice list to constrain the list of records that appear in the summary table. Users can also personalize the list of saved searches by creating new saved searches.

Review these aspects of the saved search definition process in the Oracle Fusion CRM Application Composer before you begin to define saved searches for CRM objects:

- Saved searches at runtime
- Defining a saved search

Saved Searches at Runtime

The list of saved searches is available from the **Saved Search** choice list above the overview page's summary table.

The list of saved searches includes the searches you define, as well as any personal searches that users defined themselves using personalization. Searches are displayed in alphabetical order, followed by the Personalize option.

The saved searches that you define for an object are available to all users and roles with functional security access to the object's overview page.

Defining a Saved Search

You define a saved search for an object on the object's Create Saved Search page. You define saved searches only for top-level objects, because only top-level objects have overview pages in the runtime CRM application.

To define a saved search for an object:

1. Select an application on the main Overview page.
2. Select a standard or custom object in the object tree.
3. Select the **Saved Searches** node.
4. On the Create Saved Search page for the object, enter the display label for the saved search.

The display label is the label that appears in the **Saved Search** choice list.

5. Enter the internal name and description for the saved search.

The name is automatically generated based on the specified display label, where all spaces use an underscore. For example, Top Ten Opportunities is automatically populated as Top_Ten_Opportunities.

6. Specify the search criteria that you want to include.

You can enter up to four search conditions using the following criteria:

- Specify the matching criteria, **All** or **Any**.
 - **All** appends multiple conditions with an **AND**.
 - **Any** appends two conditions with an **OR**.

If more than two conditions exist, then you cannot select **Any**.

- **Field Name**

Specify the object field whose value you want to include as a search criterion.

Long text and formula fields are not available for use in saved searches.

Tip

For better performance, the fields you include in a saved search should be indexed. You index a field when it's first created.

- **Operator**

Operator values available for selection are dependent upon the type of field selected.

This table lists, for each field type, the operators that are available for selection, as well as the values that you can enter.

Field Type	Available Operators	Available Values
<ul style="list-style-type: none"> Number Check box Percentage 	<ul style="list-style-type: none"> Equal to Greater than Greater than or equal to Is blank Is not blank Less than Less than or equal to Not equal to 	<p>Enter a literal value.</p> <hr/> <p>Note</p> <p>For percentage fields, divide the percentage by 100 to calculate the literal value you should enter as a search criterion. For example, to correctly use 150% in your saved search, you must enter 1.5.</p> <hr/>
<ul style="list-style-type: none"> Dynamic choice list Text Fixed choice list Currency 	<ul style="list-style-type: none"> Contains Does not contain Ends with Equal to Is blank Is not blank Not equal to Starts with 	<p>Enter a literal value.</p> <hr/> <p>Tip</p> <p>For fixed choice list fields, you must enter the lookup code, not the lookup meaning.</p> <hr/>
<ul style="list-style-type: none"> Date Date time 	<ul style="list-style-type: none"> After Before Equal to Is blank Is not blank Not equal to On or after On or before 	<p>Enter:</p> <ul style="list-style-type: none"> Literal value Date function <p>Specify the number of days, months, or years after or before the current date.</p>

- Value

The values that you specify are applied as the search criteria against the object records at runtime.

See the previous table.

Securing Custom Objects: Explained

After you create custom objects and fields, you then expose them on Enterprise pages for your users. Your next step is to control which users can access that object's data at runtime. By default, the object and its records are visible and editable only to a default duty role specified by the application. Grant additional access manually using the Oracle Fusion CRM Application Composer's security policy configuration pages. A security policy specifies which users are authorized to access an object's data, and what type of access they have. For example, does a user have view only access, or can the user create and update an object's record, as well? Define security policies for an object by authorizing the roles whose users can access that object's data. Or, define security policies for a role by specifying access levels across multiple custom objects.

Review these aspects of the custom object security process in the Application Composer before you begin to define your security policies:

- Securing objects
- Securing roles
- Enabling function security and data security
- CRM Application Composer and the Oracle Authorization Policy Manager (APM)
- Default security settings

Securing Objects

The object-centric Define Policies page displays a list of the enterprise-level duty roles which map to a CRM job role. Use this page to manage access to either a top-level or child custom object by specifying a security policy for one or more duty roles. When you do this, users with the corresponding roles can access the custom object and its data, depending on the security policies you define.

To access the object-centric Define Policies page:

1. Select an application on the main Overview page.
2. Select a custom object in the object tree.
3. Select the **Security** node.

Or, from the role-centric Define Policies page, select a custom object.

Roles							
Role	<input type="checkbox"/> Create	<input type="checkbox"/> View	<input type="checkbox"/> Update	<input type="checkbox"/> Delete	<input type="checkbox"/> View All	<input type="checkbox"/> Update All	<input type="checkbox"/> Grant Access
Sales Catalog Administrator Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sales Manager Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sales Administrator Duty	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Merge Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Partner Sales Manager Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Marketing Manager Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sales Analyst Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Channel Sales Manager Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sales Lead Qualifier Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CRM Application Administrator Duty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

From the object-centric Define Policies page, you can:

- Enable function security for a role.
- Enable data security for a role.
- Allow users to grant access to a specific record at runtime, to users with a given role.

Securing Roles

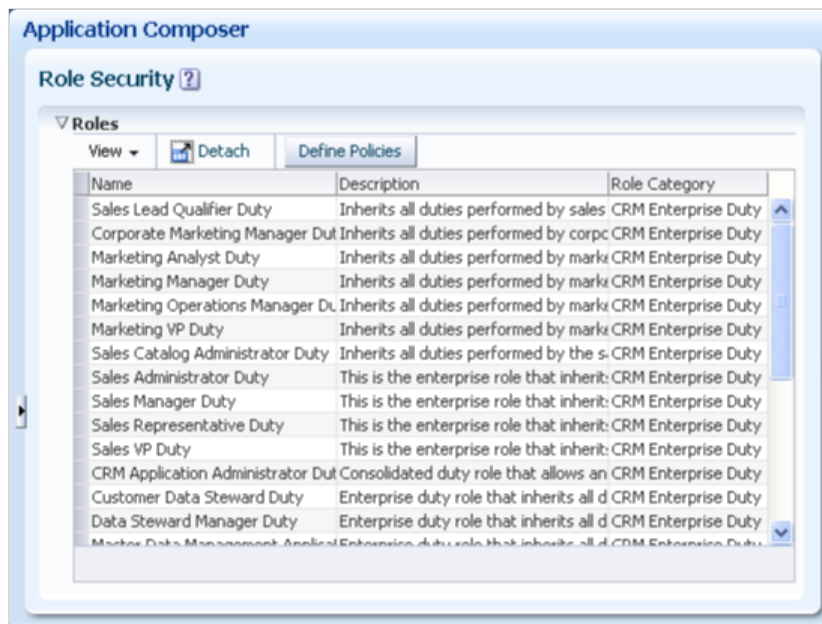
The Role Security page displays a list of the enterprise-level duty roles, which map to a CRM job role. Select a role and click the **Define Policies** button to navigate to the role-centric Define Policies page, which displays a list of the custom objects for your CRM implementation. Use this page to manage access for users with the corresponding role by specifying a security policy for one or more top-level or child custom objects. When you do this, users with the corresponding role can access the custom objects and related data, depending on the security policies you define.

To access the role-centric Define Policies page:

1. Select an application on the main Overview page.
2. Select the **Role Security** node from the Common Setup pane.

Or, select the **Role Security** hyperlink in the local area of the main Overview page.

Or, from the object-centric Define Policies page, select a role.



3. Click the **Define Policies** button.

Objects

Object	View		Update		Create		Delete		<input type="checkbox"/> View All	<input type="checkbox"/> Update All	<input type="checkbox"/> Grant Access
	<input type="checkbox"/> Access	Related Roles	<input type="checkbox"/> Access	Related Roles	<input type="checkbox"/> Access	Related Roles	<input type="checkbox"/> Access	Related Roles			
Account	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
MGTest	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto10z	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B18BAT	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Solution	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
JTestEn	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Auto10z	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
BATB18	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Activity	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tw_cust	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

From the role-centric Define Policies page, you can:

- Enable function security for a custom object.
- Enable data security for a custom object.
- Allow users to grant access to a specific record at runtime, to users with a given role.
- View related roles, if any.

If a related role is displayed next to an object, then the selected role is inheriting its access to that object from the related role. You can drill down into the related role to view its security policies.

Enabling Function Security and Data Security

A security policy specifies the type of access to an object and its records that users with the corresponding roles have. Access includes both function security as well as data security. Security settings are the same whether you are defining a security policy for an object or a role.

On the Define Policies page, the first four columns in the table manage function security, which applies to the object as a whole:

- **Create**
Users with the corresponding role can create a record of the object.
- **View**
Users with the corresponding role can view the object's work area pages.
- **Update**
Users with the corresponding role can update a record of the object.
- **Delete**
Users with the corresponding role can delete a record of the object.

The next two columns in the table manage data security.

- **View All**

Users with the corresponding role can view the object's records.

- **Update All**

Users with the corresponding role can update the object's records.

Tip

When clicking **View All** or **Update All**, the corresponding **View** and **Update** function security check boxes are automatically selected.

Wait for the page to refresh to confirm all your selections.

The last column contains the **Grant Access** check box. Selecting this check box controls the display of the **Manage Permission** button on the object's summary table at runtime. At runtime, users with the corresponding role (or an administrative role) can select an object record and click that button to specify the level of access another user should have to the selected record.

Note

To let users view or update records at runtime, you must enable both function security as well as data security for an object. To let users create records, you only have to enable function security. To let users delete records, you must grant users instance-level delete privileges using the **Grant Access** check box and the runtime **Manage Permission** button.

CRM Application Composer and the Oracle Authorization Policy Manager (APM)

Across Oracle Fusion applications, Oracle Authorization Policy Manager (APM) manages the security policies that control access based on roles. However, you define the security policies for custom objects in the Application Composer's object-centric and role-centric Define Policies pages. This is outside APM.

Since you define the security policies outside APM, you cannot later modify the security policies within APM. Instead, modify all security policies for custom objects using only the Application Composer.

Default Security Settings

By default, top-level custom objects are visible and editable only to users with a default duty role specified by a CRM application. You must manually grant additional access to other duty roles, if desired. For example, if you create a custom object in Oracle Fusion Sales, then only users with the default duty role specified by Sales are automatically granted access to that object. This lets you first access an object and its pages for testing, before you officially grant access to your customizations to users in a production environment.

This table lists the default duty roles that are provided by CRM applications:

Application	Default Duty Role
Oracle Fusion Customer Center	Sales Administrator Duty
	Marketing Operations Manager Duty
Oracle Fusion Marketing	Marketing Operations Manager Duty
Oracle Fusion Sales	Sales Administrator Duty

Child objects do not inherit security settings from parent objects. Rather, if you create a custom child object, then a default set of duty roles are granted access to the child object. In other words, a child object is visible and editable only to users with these default duty roles, as follows:

Application	Child Objects of This Parent Object	Access Granted to These Duty Roles
Oracle Fusion Customer Center	Sales Account Profile	Sales Administrator Duty Marketing Operations Manager Duty
Oracle Fusion Marketing	Sales Lead	Marketing Operations Manager Duty Sales Administrator Duty Channel Operations Manager Duty
Oracle Fusion Marketing	Marketing Campaign Marketing Response Marketing List Marketing Treatment Marketing Event Activity Marketing Advertising Activity	Marketing Operations Manager Duty
Oracle Fusion Marketing	Marketing Budget Marketing Claim Marketing Budget Entry Marketing Budget Fund Request	Marketing Operations Manager Duty Channel Operations Manager Duty
Oracle Fusion Sales	Opportunity Sales Competitor	Sales Administrator Duty
Oracle Fusion Sales	Partner	Channel Operations Manager Duty
Oracle Fusion Common CRM	Trading Community Org Contact Trading Community Resource Profile Trading Community Organization Profile Trading Community Address	Customer Data Steward Duty

Groovy Scripting: Explained

Groovy is a standard, dynamic scripting language for the Java platform for which the Oracle Fusion CRM Application Composer provides deep

support. This topic provides an overview of Groovy scripting, which you can use throughout the Application Composer to enhance your application customizations.

Before you begin to add Groovy scripts to your application customizations, read this topic, which includes:

- An explanation of terminology.
- The different contexts in the Application Composer in which you can use Groovy scripts.

Refer to the related topic, *Groovy Scripting: Examples*, to view examples of each type of Groovy script that you can write.

For more information, see the *Developer's Guide to Scripting in Oracle Fusion CRM Application Composer* on My Oracle Support at <https://support.oracle.com>. The developer's guide includes:

- The basics of Groovy.
- A compendium of tips and techniques for getting the most out of Groovy in your applications.

The tips and tricks section also includes an overview of how to work with the expression editor. The expression editor is where you write your Groovy scripts, and includes an Expression Palette, which helps you insert the names of built-in functions, object fields, or function names.

Terminology Explained

Throughout the document the term script is used to describe one or more lines of Groovy code that the Oracle ADF framework executes at runtime. Often a very-short script is all that is required.

For example, to validate that a `CommissionPercentage` field's value does not exceed 40%, you might use a one-line script like:

```
return CommissionPercentage < 0.40
```

In fact, this one-liner can be conveniently shortened by dropping the return keyword since the return keyword is always implied on the last line of a script:

```
CommissionPercentage < 0.40
```

For slightly more complicated logic, your script might require some conditional handling. For example, suppose the maximum commission percentage is 40% if the salesperson's job grade is less than or equal to 3, but 60% if the job grade is higher. Your script would grow a little to look like this:

```
if (JobGrade <= 3) {  
    return CommissionPercentage < 0.40  
}  
else {  
    return CommissionPercentage < 0.60  
}
```

Scripts that you'll write for other purposes like complex validation rules or reusable functions may span multiple pages, depending on your needs.

When a context requiring a Groovy script will typically use a short (often, one-line) script, we emphasize that fact by calling it an expression, however

technically the terms script and expression are interchangeable. Anywhere you can provide a one-line expression is also a valid context for providing a multi-line script if the need arises. Whether you provide a short expression or a multi-line script, the syntax and features at your disposal are the same. You need only pay attention that your code returns a value of the appropriate type for the context in which you use it. Each section below highlights the expected return type for the script in question.

Where You will Use Groovy in Your Application

There are a number of different contexts where you will use Groovy scripts as you customize existing objects or create new custom ones.

You will write shorter scripts to provide an expression to:

- calculate a custom formula field's value
- calculate a custom field's default value
- make a custom field conditionally updateable, or
- make a custom field conditionally required
- define the condition for executing an object workflow

You will generally write somewhat longer scripts to define:

- a field-level validation rule
- an object-level validation rule
- a trigger to complement default processing
- utility code in a global function, or
- reusable behavior in an object function

If you anticipate calling the same code from multiple different contexts, any of your scripts can call the reusable code you write in either global functions or object functions. As their name implies, global functions can be called from scripts in any object or from other global functions. Object functions can be called by any scripts in the same object, or even triggered by a button in the user interface.

For a concrete example of each of these usages, refer to the related topic, *Groovy Scripting: Examples*.

Groovy Scripting: Examples

Groovy is a standard, dynamic scripting language for the Java platform for which the Oracle Fusion CRM Application Composer provides deep support. You can use Groovy throughout the Application Composer to enhance your application customizations. This topic provides simple examples of using Groovy in all of the different supported contexts in your application.

Providing an Expression to Calculate a Custom Formula Field's Value

When you need a calculated field or a transient value-holder field with an optional initial, calculated value, use a formula field.

1. For read-only calculated fields:

A formula field defaults to being a read-only, calculated value. It displays the value resulting from the runtime evaluation of the calculation expression you supply. By using the **Depends On** multi-select list in the field create or edit page, you can configure the names of fields on which your expression depends. By doing this, its calculated value will update dynamically when any of those **Depends On** fields' value changes. The expected return type of the formula field's expression must be compatible with the formula field type you specified (Number, Date, or Text).

For example, consider a custom **TroubleTicket** object. If you add a formula field named **DaysOpen**, you can provide its calculated value with the expression:

```
(today() - CreationDate) as Integer /* truncate to whole number of days */
```

2. For transient value holder fields with optional calculated initial value:

If you want to allow the end user to override the calculated value, then mark your formula to be updateable. An updateable formula field is a "transient value holder" whose expression provides the value of the field until the user overrides it. If the user overrides the value, the object remembers this user-entered value for the duration of the current transaction so that your validation rules and triggers can reference it. If you have configured one or more **Depends On** fields for your updateable formula field, then note that the value of the formula will revert back to the calculated value should any of the dependent fields' value change. If you want a transient field whose initial value is null until the user fills it in, simply provide no formula expression for your updateable formula field to achieve this.

Providing an Expression to Calculate a Custom Field's Default Value

When a new row is created for an object, the value of a custom field defaults to **null** unless you configure a default value for it. You can supply a literal default value of appropriate type or supply an expression to calculate the default value for new rows. The default value expression is evaluated at the time the new row is created. The expected return type of your field's default value expression must be compatible with the field's type (Number, Date, Text, and so on).

For example, consider a custom **CallbackDate** field in a **TroubleTicket** object. If you want the callback back for a new trouble ticket to default to 3 days after it was created, then you can provide a default expression of:

```
CreationDate + 3
```

Providing an Expression to Make a Custom Field Conditionally Updateable

1. To provide an expression to make a custom field conditionally updateable:

A custom field can be updateable or read-only. By default, any non-formula field is updateable. Alternatively, you can configure a conditionally updateable expression. If you do this, it is evaluated each time a page displaying the field is rendered or refreshed. The expected return type the expression is **boolean**. If you define one for a field, you must also configure the **Depends On** list to indicate the names of any fields on which your conditionally updateable expression depends. By doing this, your conditionally updateable field will interactively enable or disable as appropriate when the user changes the values of fields on which the conditionally updateable expression depends.

For example, consider a custom **TroubleTicket** object with **Status** and **Justification** fields. Assume you want to prevent a user from editing the

justification of a closed trouble ticket. To achieve this, configure the conditionally updateable expression for the **Justification** field as follows:

```
Status_c != 'Closed'
```

After configuring this expression, you must then indicate that the **Justification** field depends on the **Status** field as described in "Understanding When to Configure Field Dependencies", in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer on My Oracle Support at <https://support.oracle.com>. This ensures that if a trouble ticket is closed during the current transaction, or if a closed trouble ticket is reopened, that the **Justification** field becomes enable or disabled as appropriate.

Tip

A field configured with a conditionally updateable expression only enforces the conditional updateability through the Web user interface. See "Enforcing Conditional Updateability of Custom Fields for Web Service Access" in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer, for more information on how to ensure it gets enforced for Web service access as well.

2. To provide an expression to make a custom field conditionally required:

A custom field can be optional or required. By default it is optional. Alternatively, you can configure a conditionally required expression. If you do this, it is evaluated each time a page displaying the field is rendered or refreshed, as well as when the object is validated. The expected return type of the expression **boolean**. If you define one for a field, you must also configure the **Depends On** list to indicate the names of any fields on which your conditionally required expression depends. By doing this, your conditionally required field will interactively show or hide the visual indicator of the field's being required as appropriate when the user changes the values of fields on which the conditionally required expression depends.

For example, consider a custom **TroubleTicket** object with **Priority** and **Justification** fields. Assume that priority is an integer from 1 to 5 with priority 1 being the most critical kind of problem to resolve. To enforce that a justification is required for trouble tickets whose priority is 1 or 2, configure the conditionally required expression for the **Justification** field as follows:

```
Priority_c <= 2
```

After configuring this expression, you must then indicate that the **Justification** field depends on the **Priority** field as described in "Understanding When to Configure Field Dependencies", in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer. This ensures that if a trouble ticket is created with priority 2, or an existing trouble ticket is updated to increase the priority from 3 to 2, that the **Justification** field becomes mandatory.

Defining a Field-Level Validation Rule

1. To define a field-level validation rule:

A field-level validation rule is a constraint you can define on any standard or custom field. It is evaluated whenever the corresponding field's value is set. When the rule executes, the field's value has not been assigned yet and your rule acts as a gatekeeper to its successful assignment. The expression (or longer

script) you write must return a **boolean** value that indicates whether the value is valid. If the rule returns **true**, then the field assignment will succeed so long as all other field-level rules on the same field also return **true**. If the rule returns **false**, then this prevents the field assignment from occurring, the invalid field is visually highlighted in the UI, and the configured error message is displayed to the end user. Since the assignment fails in this situation, the field retains its current value (possibly **null**, if the value was **null** before), however the UI component in the Web page allows the user to see and correct their invalid entry to try again. Your script can use the **newValue** keyword to reference the new value that will be assigned if validation passes. To reference the existing field value, use the **oldValue** keyword. A field-level rule is appropriate when the rule to enforce only depends on the new value being set. You can use the Keywords tab of the Expression Palette to insert the **newValue** and **oldValue** keywords.

For example, consider a custom **TroubleTicket** object with a **Priority** field. To validate that the number entered is between 1 and 5, your field-level validation rule would look like this:

Field-Level Validation Rule Component	Value
Field Name	Priority
Rule Name	Validate_Priority_Range
Rule Body	<code>newValue == null (1..5).contains(newValue as Integer)</code>
Error Message	The priority must be in the range from 1 to 5.

Tip

If a validation rule for field **A** depends on the values of one or more other fields (e.g. **Y** and **Z**), then create an object-level rule and programmatically signal which field or fields should be highlighted as invalid to the user, as explained in "Setting Invalid Fields for the UI in an Object-Level Validation Rule" in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer.

2. To define an object-level validation rule:

An object-level validation rule is a constraint you can define on any standard or custom object. It is evaluated whenever the framework attempts to validate the object. This can occur upon submitting changes in a web form, when navigating from one row to another, as well as when changes to an object are saved. Use object-level rules to enforce conditions that depend on two or more fields in the object. This ensures that regardless of the order in which the user assigns the values, the rule will be consistently enforced. The expression (or longer script) you write must return a **boolean** value that indicates whether the object is valid. If the rule returns **true**, then the object validation will succeed so long as all other object-level rules on the same object return **true**. If the rule returns **false**, then this prevents the object from being saved, and the configured error message is displayed to the end user.

For example, consider a **TroubleTicket** object with **Priority** and **AssignedTo** fields, where the latter is a dynamic choice list field referencing **Contact** objects whose **Type** field is a **Staff Member**. To validate that a trouble ticket of priority 1 or 2 cannot be saved without being assigned to a staff member, your object-level rule would look like this:

Object-Level Validation Rule Component	Value
Rule Name	Validate_High_Priority_Ticket_Has_Owner
Rule Body	<pre>// Rule depends on two fields, so must be // written as object-level rule if (Priority_c <= 2 && AssignedTo_Id_c == null) { // Signal to highlight the AssignedTo field // on the UI as being in error adf.error.addAttribute('AssignedTo_c') return false; } return true;</pre>
Error Message	A trouble ticket of priority 1 or 2 must have a staff member assigned to it.

Defining Utility Code in a Global Function

Global functions are useful for code that multiple objects want to share. To call a global function, preface the function name with the `adf.util.` prefix. When defining a function, you specify a return value and can optionally specify one or more typed parameters that the caller will be required to pass in when invoked. The most common types for function return values and parameters are the following:

- **String:** a text value
- **Boolean:** a logical true or false value
- **Long:** an integer value
- **BigInteger:** a integer of arbitrary precision
- **Double:** a floating-point decimal value
- **BigDecimal:** a decimal number of arbitrary precision
- **Date:** a date value with optional time component
- **List:** an ordered collection of objects
- **Map:** an unordered collection of name/value pairs
- **Object:** any object

In addition, a function can define a **void** return type which indicates that it returns no value.

Note

.A global function has no current object context. To write global functions that work on a particular object, refer to "Passing the Current Object to a Global Function," in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer.

For example, the following two global functions define standard helper routines to log the start of a block of Groovy script and to log a diagnostic message. Examples in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer make use of them.

This table describes how to set up a global function to log the start of a block of Groovy script.

Global Function Component	Value
Function Name	logStart
Return Type	void
Parameters	<ul style="list-style-type: none"> • Name: scriptName • Type: String
Function Definition	<pre>// Log the name of the script println("[In: \${scriptName}]")</pre>

This table describes how to set up a global function to log a diagnostic message.

Global Function Component	Value
Function Name	log
Return Type	void
Parameters	<ul style="list-style-type: none"> • Name: message • Type: String
Function Definition	<pre>// Log the message, could add other info println(message)</pre>

Defining Reusable Behavior with an Object Function

Object functions are useful for code that encapsulates business logic specific to a given object. You can call object functions by name from any other script code related to the same object. In addition, you can invoke them using a button or link in the user interface. The supported return types and optional parameter types are the same as for global functions (described above).

For example, you might define the following **updateOpenTroubleTicketCount()** object function on a **Contact** custom object. It begins by calling the **logStart()** global function above to log a diagnostic message in a standard format to signal the beginning of a block of custom Groovy script. It calls the **newView()** built-in function (described in "Accessing the View Object for Programmatic Access to Business Objects" in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer) to access the view object for programmatic access of trouble tickets, then calls another global function **applyFilter()** (described in "Simplifying Most Common View Criteria Creation with a Global Function" in the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer) to apply a filter to find trouble tickets related to the current contact's id and having either **Working** or **Waiting** as their current status. Finally, it calls **getEstimatedRowCount()** to retrieve the count of trouble tickets that qualify for the filter criteria.

This table describes the **updateOpenTroubleTicketCount()** object function on a **Contact** custom object:

Object Function Component	Value
Function Name	updateOpenTroubleTicketCount

Return Type	void
Parameters	None
Function Definition	<pre> adf.util.logStart('updateOpenTroubleTicketCount') // Access the view object for TroubleTicket // programmatic access def tickets = newView('TroubleTicket_c') // Use a global function to apply a query // filter to Trouble Ticket // view object where Contact_Id = Id of // current Contact and // Status is either 'Working' or 'Waiting' adf.util.applyFilter(tickets,[Status_c: ['Working','Waiting'], Contact_Id__c:Id]) // Update OpenTroubleTickets field value setAttribute('OpenTroubleTickets_c',tickets.getEstimatedRowCount) </pre>

Defining a Trigger to Complement Default Processing

Triggers are scripts that you can write to complement the default processing logic for a standard or custom object. The following triggers are available:

- **After Create:** Fires when a new instance of an object is created. Use to assign programmatic default values to one or more fields in the object.
- **Before Modify:** Fires when the end-user first modifies a persistent field in an existing, queried row.
- **Before Invalidate:** Fires on an existing object when its first persistent field is modified. Fires on a valid parent object when a child row is created, removed, or modified.
- **Before Remove:** Fires when an attempt is made to delete an object. Returning false stops the row from being deleted and displays the optional trigger error message.
- **Before Insert in Database:** Fires before a new object is inserted into the database.
- **After Insert in Database:** Fires after a new object is inserted into the database.
- **Before Update in Database:** Fires before an existing object is modified in the database.
- **After Update in Database:** Fires after an existing object is modified in the database.
- **Before Delete in Database:** Fires before an existing object is deleted from the database.
- **After Delete in Database:** Fires after an existing object is deleted from the database.
- **Before Commit in Database:** Fires before the change pending for the current object (insert, update, delete) is made permanent in the current transaction.
- **After Commit in Database:** Fires after the change pending for the current object (insert, update, delete) is made permanent in the current transaction.

- **Before Rollback in Database:** Fires before the change pending for the current object (insert, update, delete) is rolled back.
- **After Rollback in Database:** Fires after the change pending for the current object (insert, update, delete) is rolled back.

For example, consider a Contact object with a **OpenTroubleTickets** field that needs to be updated any time a trouble ticket is created or modified. You can create the following two triggers on the **TroubleTicket** object which both invoke the **updateOpenTroubleTicketCount()** object function described above.

This table describes how to set up a trigger on the **TroubleTicket** object.

Trigger Component	Value
Trigger Object	TroubleTicket
Trigger	After Insert In Database
Trigger Name	After_Insert_Set_Open_Trouble_Tickets
Trigger Definition	<pre> adf.util.logStart('After_Insert_Set_Open_Trouble_Tickets') // Get the related contact for this trouble ticket def relatedContact = Contact_Obj_c // Update its OpenTroubleTickets field value relatedContact?.updateOpenTroubleTicketCount()</pre>

This table describes how to set up a second trigger on the **TroubleTicket** object.

Trigger Component	Value
Trigger Object	TroubleTicket
Trigger	After Update In Database
Trigger Name	After_Update_Set_Open_Trouble_Tickets
Trigger Definition	<pre> // Get the related contact for this trouble ticket def relatedContact = Contact_Obj_c // Update its OpenTroubleTickets field value relatedContact?.updateOpenTroubleTicketCount()</pre>

Supported Classes and Methods for Use in Groovy Scripts: Explained

Groovy is a standard, dynamic scripting language for the Java platform for which the Oracle Fusion CRM Application Composer provides deep support. This topic covers the supported classes and methods for use in Groovy scripts.

Classes and Methods

When writing Groovy scripts, you may only use the classes and methods that are documented in the table below. Using any other class or method may work initially, but will throw a runtime exception when you migrate your code to later versions. Therefore, we strongly suggest that you ensure the Groovy code you write adheres to the classes and methods shown here. For each class, in addition to the method names listed in the table, the following method names are also allowed:

- equals()
- hashCode()
- toString()

In contrast, the following methods are never allowed on any object:

- finalize()
- getClass()
- getMetaClass()
- notify()
- notifyAll()
- wait()

Class Name	Allowed Methods	Package
ADFContext	getLocale() getSecurityContext() getUserRoles() isUserInRole()	oracle.adf.share
Array	Any constructor Any method	java.sql
Array	getArray() getElemType() getList()	oracle.jbo.domain
ArrayList	Any constructor Any method	java.util
Arrays	Any constructor Any method	java.util
AttributeDef	getAttributeKind() getIndex() getJavaType() getName() getPrecision() getProperty() getScale() getUIHelper() getUpdateableFlag() isMandatory() isQueryable()	oracle.jbo

AttributeHints	getControlType() getDisplayHeight() getDisplayHint() getDisplayWidth() getFormat() getFormattedAttribute() getFormatter() getFormatterClassName() getHint() getLocaleName() parseFormattedAttribute()	oracle.jbo
AttributeList	getAttribute() getAttributeIndexOf() getAttributeNames() setAttribute()	oracle.jbo
BaseLobDomain	closeCharacterStream() closeInputStream() closeOutputStream() getInputStream() getLength() getOutputStream() getcharacterStream()	oracle.jbo.domain
BigDecimal	Any constructor Any method	java.math
BigInteger	Any constructor Any method	java.math
BitSet	Any constructor Any method	java.util
Blob	Any constructor Any method	java.sql
BlobDomain	Any constructor getBinaryOutputStream() getBinaryStream() getBufferSize()	oracle.jbo.domain

Boolean	Any constructor	java.lang
	Any method	
Byte	Any constructor	java.lang
	Any method	
Calendar	Any constructor	java.util
	Any method	
Char	Any constructor	oracle.jbo.domain
	bigDecimalValue()	
	bigIntegerValue()	
	booleanValue()	
	doubleValue()	
	floatValue()	
	getValue()	
	intValue()	
	longValue()	
Clob	Any constructor	java.sql
	Any method	
ClobDomain	Any constructor	oracle.jbo.domain
	toCharArray()	
Collection	Any constructor	java.util
	Any method	
Comparator	Any constructor	java.util
	Any method	
Currency	Any constructor	java.util
	Any method	
DBSequence	Any constructor	oracle.jbo.domain
	getValue()	
Date	Any constructor	java.util
	Any method	
Date	Any constructor	java.sql
	Any method	

Date	Any constructor compareTo() dateValue() getValue() stringValue() timeValue() timestampValue()	oracle.jbo.domain
Dictionary	Any constructor Any method	java.util
Double	Any constructor Any method	java.lang
Enum	Any constructor Any method	java.lang
EnumMap	Any constructor Any method	java.util
EnumSet	Any constructor Any method	java.util
Enumeration	Any constructor Any method	java.util
EventListener	Any constructor Any method	java.util
EventListenerProxy	Any constructor Any method	java.util
EventObject	Any constructor Any method	java.util
Exception	Any constructor Any method	java.lang
ExprValueErrorHandler	addAttribute() clearAttributes() raise() raiseLater() warn()	oracle.jbo
Float	Any constructor Any method	java.lang

Formattable	Any constructor	java.util
	Any method	
FormattableFlags	Any constructor	java.util
	Any method	
Formatter	Any constructor	java.util
	Any method	
GregorianCalendar	Any constructor	java.util
	Any method	
HashMap	Any constructor	java.util
	Any method	
HashSet	Any constructor	java.util
	Any method	
Hashtable	Any constructor	java.util
	Any method	
IdentityHashMap	Any constructor	java.util
	Any method	
Integer	Any constructor	java.lang
	Any method	
Iterator	Any constructor	java.util
	Any method	
JboException	getDetails() getErrorCode() getErrorParameters() getLocalizedMessage() getMessage() getProductCode() getProperty()	oracle.jbo
JboWarning	Any constructor getDetails() getErrorCode() getErrorParameters() getLocalizedMessage() getMessage() getProductCode() getProperty()	oracle.jbo

Key	toStringFormat()	oracle.jbo
LinkedHashMap	Any constructor	java.util
	Any method	
LinkedHashSet	Any constructor	java.util
	Any method	
LinkedList	Any constructor	java.util
	Any method	
List	Any constructor	java.util
	Any method	
ListIterator	Any constructor	java.util
	Any method	
ListResourceBundle	Any constructor	java.util
	Any method	
Locale	Any constructor	java.util
	Any method	
Long	Any constructor	java.lang
	Any method	
Map	Any constructor	java.util
	Any method	
Math	Any constructor	java.lang
	Any method	
MathContext	Any constructor	java.math
	Any method	
NClob	Any constructor	java.sql
	Any method	
NameValuePair	Any constructor	oracle.jbo
	getAttribute()	
	getAttributeIndexOf()	
	getAttributeNames()	
NativeTypeDomainInterface	setAttribute()	oracle.jbo.domain
	getNativeObject()	

Number	Any constructor bigDecimalValue() bigIntegerValue() booleanValue() byteValue() doubleValue() floatValue() getValue() intValue() longValue() shortValue()	oracle.jbo.domain
Observable	Any constructor Any method	java.util
Observer	Any constructor Any method	java.util
PriorityQueue	Any constructor Any method	java.util
Properties	Any constructor Any method	java.util
PropertyPermission	Any constructor Any method	java.util
PropertyResourceBundle	Any constructor Any method	java.util
Queue	Any constructor Any method	java.util
Random	Any constructor Any method	java.util
RandomAccess	Any constructor Any method	java.util
Ref	Any constructor Any method	java.sql
ResourceBundle	Any constructor Any method	java.util

Row	getAttribute() getAttributeHints() getKey() getLookupDescription() getOriginalAttribute() getOriginalAttributeValue() isAttributeChanged() isAttributeUpdateable() remove() revertRow() revertRowAndContainees() setAttribute() validate()	oracle.jbo
RowId	Any constructor Any method	java.sql
RowIterator	createAndInitRow() createRow() findByKey() findRowsMatchingCriteria() first() getAllRowsInRange() getCurrentRow() insertRow() last() next() previous() reset()	oracle.jbo

RowSet	createAndInitRow() createRow() executeQuery() findByKey() findRowsMatchingCriteria() first() getAllRowsInRange() getCurrentRow() insertRow() last() next() previous() reset()	oracle.jbo
Scanner	Any constructor Any method	java.util
SecurityContext	getUsername() getUserProfile()	oracle.adf.share.security
Session	getLocale() getLocaleContext() getUserData()	oracle.jbo
Set	Any constructor Any method	java.util
Short	Any constructor Any method	java.lang
Short	Any constructor Any method	java.lang
SimpleTimeZone	Any constructor Any method	java.util
SortedMap	Any constructor Any method	java.util
SortedSet	Any constructor Any method	java.util
Stack	Any constructor Any method	java.util

StackTraceElement	Any constructor	java.lang
	Any method	
StrictMath	Any constructor	java.lang
	Any method	
String	Any constructor	java.lang
	Any method	
StringBuffer	Any constructor	java.lang
	Any method	
StringBuilder	Any constructor	java.lang
	Any method	
StringTokenizer	Any constructor	java.util
	Any method	
Struct	Any constructor	java.sql
	Any method	
Struct	getAttribute()	oracle.jbo.domain
	setAttribute()	
StructureDef	findAttributeDef()	oracle.jbo
	getAttributeIndexOf()	
Time	Any constructor	java.sql
	Any method	
TimeZone	Any constructor	java.util
	Any method	
Timer	Any constructor	java.util
	Any method	
TimerTask	Any constructor	java.util
	Any method	
Timestamp	Any constructor	java.sql
	Any method	
Timestamp	Any constructor	oracle.jbo.domain
	compareTo()	
	dateValue()	
	getValue()	
	stringValue()	
	timeValue()	
	timestampValue()	

TreeMap	Any constructor	java.util
	Any method	
TreeSet	Any constructor	java.util
	Any method	
UUID	Any constructor	java.util
	Any method	

UserProfile	getBusinessCity() getBusinessCountry() getBusinessEmail() getBusinessFax() getBusinessMobile() getBusinessPOBox() getBusinessPager() getBusinessPhone() getBusinessPostalAddr() getBusinessPostalCode() getBusinessState() getBusinessStreet() getDateOfBirth() getDateofHire() getDefaultGroup() getDepartment() getDepartmentNumber() getDescription() getDisplayName() getEmployeeNumber() getEmployeeType() getFirstName() getGUID() getGivenName() getHomeAddress() getHomePhone() getInitials() getJpegPhoto() getLastName() getMaidenName() getManager() getMiddleName() getName() getNameSuffix() getOrganization() getOrganizationalUnit() getPreferredLanguage()	oracle.adf.share.security.identitymanagement
-------------	--	--

ValidationException	getDetails() getErrorCode() getErrorParameters() getLocalizedMessage() getMessage() getProductCode() getProperty()	oracle.jbo
Vector	Any constructor Any method	java.util
ViewCriteria	createAndInitRow() createRow() createViewCriteriaRow() findByKey() findRowsMatchingCriteria() first() getAllRowsInRange() getCurrentRow() insertRow() last() next() previous() reset()	oracle.jbo
ViewCriteriaItem	getValue() makeCompound() setOperator() setUpperColumns() setValue()	oracle.jbo
ViewCriteriaItemCompound	ensureItem() getValue() makeCompound() setOperator() setUpperColumns() setValue()	oracle.jbo

ViewCriteriaRow	ensureCriteriaItem() getConjunction() isUpperColumns() setConjunction() setUpperColumns()	oracle.jbo
ViewObject	appendViewCriteria() createAndInitRow() createRow() createViewCriteria() executeQuery() findByKey() findRowsMatchingCriteria() first() getAllRowsInRange() getCurrentRow() getMaxFetchSize() insertRow() last() next() previous() reset() setMaxFetchSize()	oracle.jbo
WeakHashMap	Any constructor Any method	java.util

Importing and Exporting Custom Objects: Explained

To support the importing and exporting of the custom objects that you created with the Oracle Fusion CRM Application Composer, you must first generate the object artifacts required for both file-based import and bulk export.

Oracle Fusion Import and Export Processes

In Oracle Fusion, two processes exist to enable the importing and exporting of object data: file-based import and bulk export.

File-based import supports the import of data from an external text or xml file to interface tables and then from interface tables to target application tables.

Note

File-based import bypasses any Groovy validation and trigger logic on an object.

Bulk export lets you extract large volumes of data from CRM objects, both as extracts of a full set of records for an object as well as incremental extracts. The system creates comma or tab-delimited files with the extracted data, which are available to users as attachments to the batch records that have been executed.

Enabling Import and Export for Custom Objects

The object model extensions that you make using the Application Composer do not create the artifacts required by these import and export processes. For example, file-based import leverages Oracle Data Integrator (ODI).

Accordingly, after completing your object model extensions, generate the required artifacts to register your extensions and make them available for importing and exporting.

Note

The creation of import and export artifacts occurs only in the Oracle Metadata Services (MDS) mainline, and is not supported within the MDS sandbox.

To enable the import and export of custom object data:

1. Select an application on the main Overview page.
2. Select the **Import and Export** link in the Common Setup pane, or in the local area of the main Overview page.
3. On the Import and Export page, click the **Generate** button.

After you enable your object model extensions for importing and exporting, you can then schedule your file-based import and bulk export processes using the following tasks, available by selecting **Setup and Maintenance** from the Tools menu and searching on the task name.

- To schedule your custom object imports, select the **Manage File Import Activities** task.

To initially set up file-based import for importing custom object data, select the **Manage File Import Objects** and **Manage File Import Mappings** tasks.

Note

Custom child objects are imported as part of the parent object.

- To schedule your custom object exports, select the **Schedule Export Processes** task.
-

Note

Both top-level and child custom objects are available as independent exportable objects.

Important

Refer to Oracle Fusion CRM product-specific documentation for additional details on how Oracle Fusion CRM products enable the import and export of custom object data (custom fields) for standard objects.

FAQs for Extending Oracle Fusion CRM Applications

What's the difference between fixed choice lists and dynamic choice lists?

A fixed choice list and a dynamic choice list are similar in that the ultimate goal of both types of choice lists is to generate a field with a list of values at runtime. However, the list of values for a fixed choice list is derived from an FND lookup type.

The list of values for a dynamic choice list is derived from an existing object's actual data.

What's the difference between Oracle Composer and Oracle Fusion CRM Application Composer?

Oracle Composer is an Oracle Fusion tool you can use to modify Oracle Fusion user interface (UI) pages and components for all products designated for use with Oracle Composer. Oracle Composer uses two different modes of Design View. The first mode, Design View: Standard mode, is selected by default in all CRM pages when opening a page with Oracle Composer with the Design button selected. The second mode, Design View: Direct Selection mode, is activated when you click the Select tab for the UI page you want to customize. In CRM, Direct Selection mode is available when you customize pages, but not when you personalize a dashboard page. With the Design View: Direct Selection mode, you can select and edit UI elements such as form fields and table columns. In Direct Selection mode, selectable UI components become apparent when you move your cursor over the UI component. Selectable UI components are highlighted and can be edited.

This table describes how you can use each mode of Oracle Composer to customize dashboard pages and other select pages (such as the Partner Public Profile page, Partner Landing page, Partner Registration, Customer Snapshot, and Customer Overview - Analysis tab), and customize transactional pages (all other non-dashboard pages):

Use Cases	Design View - Standard mode	Design View - Direct Selection mode	Page Type
Add content (Business Intelligence reports, CRM portlets such as Calendar)	Yes	No	Dashboard and other select pages
Delete region	Yes	No	Dashboard and other select pages
Move region	Yes	No	Dashboard and other select pages

Change page layout (for example, change a two column layout to three column layout)	Yes	No	Dashboard and other select pages
Default region state (open or close)	Yes	No	Transactional pages (all non-dashboard pages)
Manage save queries (create and edit)	Yes	No	Transactional pages (all non-dashboard pages)
Hide or show field	No	Yes	Transactional pages (all non-dashboard pages)
Change field label	No	Yes	Transactional pages (all non-dashboard pages)
Make field required or not	No	Yes	Transactional pages (all non-dashboard pages)
Make field read-only or updateable	No	Yes	Transactional pages (all non-dashboard pages)
Reorder fields in a Form	No	Yes	Transactional pages (all non-dashboard pages)
Reorder table columns	Yes	Yes	Transactional pages (all non-dashboard pages)
Hide or show table columns	Yes	Yes	Transactional pages (all non-dashboard pages)
Set table column width with the mouse	Yes	No	Transactional pages (all non-dashboard pages)
Set table column width and min width in percent or pixels	No	Yes	Transactional pages (all non-dashboard pages)
Make column sortable or not	No	Yes	Transactional pages (all non-dashboard pages)

The Oracle Fusion CRM Application Composer also lets you make UI changes at runtime. However, the types of UI changes that you can make using the Application Composer are quite different. Specifically, your primary focus when using the Application Composer is to make actual object model changes. For example, you can create a new business object and related fields, and then create new application pages where that object and its fields are exposed to users. The ability to make these types of object model extensions is available only in Oracle Fusion CRM applications. Also, using the Application Composer, you cannot access the Resource Catalog to add new content to a page. With Application Composer, administrators can make customizations at the site level only.

This table describes some of the primary differences between Oracle Composer and the Application Composer:

Customization Task	Available in Oracle Composer (site, job role, external or internal level)?	Available in Oracle Fusion CRM Application Composer (site level only)?
Make object model extensions and expose your customizations by creating or modifying work area pages	No	Yes

Reorder subtabs	No	Yes
Customize dashboard pages	Yes	No
Add content from the Resource Catalog	Yes	No
Simple field customizations (show, hide, make read only, make required)	Yes (WYSIWYG - what you see is what you get)	Yes (non-WYSIWYG)
Select the MDS layer where you want to author customizations, such as at the site layer or job role layer	Yes	No
View results of customizations immediately	Yes, in the Oracle Composer design interface	Yes, in the CRM application that you are customizing

Application Composer: Extending Oracle Fusion CRM Applications

Customizing Oracle Fusion CRM Applications Using Oracle Fusion CRM Application Composer: Explained

The Oracle Fusion CRM Application Composer provides a series of task flows which let you customize and extend an Oracle Fusion CRM application according to the needs of your users. For example, you can create fields for an existing standard object, and expose those new fields on the object's work area. Or, create a brand new custom object and related fields, then create a work area where that object and its fields are exposed to users. The task flows available to you are dependent upon the CRM application that you are customizing. This topic provides an overview of which CRM Application Composer task flows are available for use in each CRM application.

This topic addresses extensibility for these CRM applications:

- Oracle Fusion Common CRM
- Oracle Fusion Customer Center
- Oracle Fusion Marketing
- Oracle Fusion Sales
- Oracle Fusion Sales Catalog

You can also refer to the product-specific implementation guides to learn more about how a particular application works with the Application Composer.

Oracle Fusion Common CRM

Using the Common application container in the Application Composer, you can complete the tasks listed in the tables below.

This includes Master Data Management (MDM) and Common Party User Interface (CPUI) objects.

Note

You cannot create custom objects using the Common application container in the Application Composer.

Customization Task Flow	Task	Interaction	Note
Create or Edit Top Level Object			
Create or Edit Child Object for Standard and Custom Parent Object	No	No	No
Create or Edit Text Attribute	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	Yes	Yes
Create or Edit Date-time Attribute	No	Yes	Yes
Create or Edit Long-text Attribute	No	Yes	Yes
Create or Edit Check-box Attribute	No	Yes	Yes
Create or Edit Percentage Attribute	No	Yes	Yes
Create or Edit Currency Attribute	No	Yes	Yes
Create or Edit Fixed Choice List Attribute	No	Yes	Yes
Create or Edit Dynamic Choice List Attribute	No	Yes	Yes
Create or Edit Formula Attribute	No	Yes	Yes
Create Indexed Attribute	No	Yes	Yes
Create or Edit Action with Script (using groovy)	No	Yes	Yes
Create or Edit Action with URL	No	Yes	Yes
Create or Edit Links with URL	No	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	No	Yes	Yes
Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	No	Yes	Yes

Create, Edit or Delete Triggers using groovy scripts	No	Yes	Yes
Create, Edit or Delete Object Functions using groovy scripts	No	Yes	Yes
Create or Edit Saved Search for a Parent Object	No	Yes	Yes
Create Work Area for a Custom Top Level Object			
Add, Edit Custom or Standard fields to Local Search	No	No	No
Add, Edit Custom or Standard fields to Regional Search	No	No	No
Add or Edit Custom or Standard fields to Summary Table	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Creation Page	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Summary Form	Yes	Yes	Yes
Add Actions or Links to Summary Table	No	Yes	Yes
Add Actions or Links to Summary Form	No	Yes	Yes
Create or Edit Picker	No	No	No
Create or Edit Regional Pane	No	No	No
Create or Edit Tree Nodes	No	No	No
Create or Edit Child or Related Object subtab	No	No	No
Create or Edit Context Link subtab	No	No	No
Create or Edit Common Component subtab	No	No	No
Create or Edit Web Content subtab	No	No	No
Create Mobile Pages	No	No	No
Create Tablet Pages	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom text attribute	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Number attribute	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	No	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	No	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	No	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	No	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	No	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	No	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	No	Yes	Yes
Find and get a record Standard or Custom object having custom attribute	No	Yes	Yes
Create an action script that triggers a Web service	No	Yes	Yes

Create or Edit Object workflow	No	Yes	Yes
Create or Edit or Delete Field Updates	No	Yes	Yes
Create or Edit or Delete E-mail Notification	No	Yes	Yes
Create or Edit or Delete Outbound Message	No	Yes	Yes
Create or Edit or Delete Business-process Flow	No	Yes	Yes
Create or Edit or Delete Task Creation	No	No	No
Execute Object Workflow for all actions	No	Yes	Yes
Create or Edit or Delete E-mail Templates	No	Yes	Yes
Create or Edit a Relationship	No	No	No

Table (continued from above): Master Data Management (MDM) and Common Party User Interface (CPUI) objects

Customizati Task Flow	Trading Community Organizatio Contact	Trading Community Resource Profile	Trading Community Organizatio Profile	Trading Community Person Profile	Trading Community Relationship	Trading Community Address	Trading Community Group Profile
Create or Edit Top Level Object							Not applicable
Create or Edit Child Object for Standard and Custom Parent Object	Yes	Yes	No	No	No	Yes	Not applicable
Create or Edit Text Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Date-time Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Create or Edit Long-text Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Check-box Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Percentage Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Currency Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Fixed Choice List Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Dynamic Choice List Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Formula Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create Indexed Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with Script (using groovy)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with URL	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Links with URL	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Functions using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Saved Search for a Parent Object	No	Yes	No	No	No	No	Yes
Create, Update or Delete roles for Custom Object							Not applicable
Create, Update or Delete permissions for Custom Object							Not applicable
View APM roles defined across the applications for Custom Object							Not applicable
View or Edit policies defined for a role (Application Specific Custom Objects)							Not applicable

Create Work Area for a Custom Top Level Object							Not applicable
Add, Edit Custom or Standard fields to Local Search	No	Yes	No	No	No	No	Yes
Add, Edit Custom or Standard fields to Regional Search	No	No	No	No	No	No	No
Add, Edit Custom or Standard fields to Summary Table	Yes	Yes	No	No	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Creation Page	Yes	No	Yes	Yes	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Summary Form	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Add Actions or Links to Summary Table	Yes	Yes	No	No	Yes	Yes	Yes
Add Actions or Links to Summary Form	Yes	Yes	Yes	Yes	No	No	Yes
Create or Edit Picker	No	No	No	No	No	No	No
Create or Edit Regional Pane	No	No	No	No	No	No	No
Create or Edit Tree Nodes	No	No	No	No	No	No	No

Create or Edit Child or Related Object subtab	No	Yes	No	No	No	No	No
Create or Edit Context Link subtab	No	Yes	No	No	No	No	No
Create or Edit Common Component subtab	No	No	No	No	No	No	No
Create or Edit Web Content subtab	No	Yes	No	No	No	No	No
Create Mobile Pages	Yes	No	No	No	No	No	No
Create Tablet Pages	No	No	No	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom text attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Number attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Find and get a record Standard or Custom object having custom attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create an action script that triggers a Web service	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Save and Publish a Subject Area	No	Yes	Yes	Yes	No	No	No
Edit or Save republish a Subject Area	No	Yes	Yes	Yes	No	No	No
Create or Edit Object workflow	Yes	Yes	No	No	Yes	Yes	No
Create or Edit or Delete Field Updates	Yes	Yes	No	No	Yes	Yes	No

Create or Edit or Delete E-mail Notification	Yes	Yes	No	No	Yes	Yes	No
Create or Edit or Delete Outbound Message	Yes	Yes	No	No	Yes	Yes	No
Create or Edit or Delete Business-process Flow	Yes	Yes	No	No	Yes	Yes	No
Create or Edit or Delete Task Creation	No	No	No	No	No	No	No
Execute Object Workflow for all actions	Yes	Yes	No	No	Yes	Yes	No
Create or Edit or Delete E-mail Templates	Yes	Yes	No	No	Yes	Yes	No
Create or Edit a Relationship	Yes	Yes	No	No	No	Yes	No
Create, Edit, Delete Global Functions using groovy scripts							Yes
Generate Import and Export artifacts							Yes
Enable and Disable Application Script Logging							Yes

Oracle Fusion Customer Center

Using the Customer Center application container in the Application Composer, you can:

- Create custom objects
- Complete the tasks listed in the table below for standard objects.

Customization Task Flow	Custom Object	Sales Account Resource	Sales Account
Create or Edit Top Level Object	Yes		
Create or Edit Child Object for Standard and Custom Parent Object	Yes	No	Yes
Create or Edit Text Attribute	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	Yes	Yes
Create or Edit Date-time Attribute	Yes	Yes	Yes
Create or Edit Long-text Attribute	Yes	Yes	Yes
Create or Edit Check-box Attribute	Yes	Yes	Yes
Create or Edit Percentage Attribute	Yes	Yes	Yes
Create or Edit Currency Attribute	Yes	Yes	Yes
Create or Edit Fixed Choice List Attribute	Yes	Yes	Yes
Create or Edit Dynamic Choice List Attribute	Yes	Yes	Yes
Create or Edit Formula Attribute	Yes	Yes	Yes
Create Indexed Attribute	Yes	Yes	Yes
Create or Edit Action with Script (using groovy)	Yes	Yes	Yes
Create or Edit Action with URL	Yes	Yes	Yes
Create or Edit Links with URL	Yes	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes
Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes	Yes	Yes

Create, Edit or Delete Object Functions using groovy scripts	Yes	Yes	Yes
Create or Edit Saved Search for a Parent Object			
Add, Edit Custom or Standard fields to Local Search	Yes	No	No
Add, Edit Custom or Standard fields to Regional Search	Yes	No	No
Add or Edit Custom or Standard fields to Summary Table	Yes	No	No
Add or Edit Custom or Standard fields to Creation Page	Yes	No	No
Add or Edit Custom or Standard fields to Summary Form	Yes	No	Yes
Add Actions or Links to Summary Table	Yes	No	Yes
Add Actions or Links to Summary Form		No	
Create or Edit Picker		No	
Create or Edit Regional Pane		No	Yes
Create or Edit Tree Nodes	No	No	Yes
Create or Edit Child or Related Object subtab	Yes	No	
Create or Edit Context Link subtab	Yes	No	
Create or Edit Common Component subtab	Yes	No	
Create or Edit Web Content subtab	Yes	No	
Create Mobile Pages		No	
Create Tablet Pages		No	
Create, Update and Delete a record in Standard or Custom object having custom text attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Number attribute	Yes	No	Yes

Create, Update and Delete a record in Standard or Custom object having custom Date attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	Yes	No	Yes
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	Yes	No	Yes
Find and get a record Standard or Custom object having custom attribute	Yes	No	Yes
Create an action script that triggers a Web service			
Create or Edit or Save and Publish a Subject Area		No	
Edit or Save republish a Subject Area		No	

Create or Edit Object workflow	Yes	Yes	Yes
Create or Edit or Delete Field Updates	Yes	Yes	Yes
Create or Edit or Delete E-mail Notification	Yes	Yes	Yes
Create or Edit or Delete Outbound Message	Yes	Yes	Yes
Create or Edit or Delete Business-process Flow	Yes	Yes	Yes
Create or Edit or Delete Task Creation	Yes	Yes	Yes
Execute Object Workflow for all actions	Yes	Yes	Yes
Create or Edit or Delete E-mail Templates	Yes	Yes	Yes
Create or Edit a Relationship	Yes	No	Yes

Oracle Fusion Marketing

Using the Marketing application container in the Application Composer, you can:

- Create custom objects
- Complete the tasks listed in the tables below for standard objects.

Customization Task Flow	Custom Object	Sales Lead Version 2	Marketing Campaign	Marketing Budget Version 2	Marketing List	Marketing Treatment
Create or Edit Top Level Object	Yes					
Create or Edit Child Object for Standard and Custom Parent Object	Yes	Yes	No	Yes	No	No
Create or Edit Text Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Date-time Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Long-text Attribute	Yes	Yes	Yes	Yes	Yes	Yes

Create or Edit Check-box Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Percentage Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Currency Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Fixed Choice List Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Dynamic Choice List Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Formula Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create Indexed Attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with Script (using groovy)	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with URL	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Links with URL	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes

Create, Edit or Delete Object Functions using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Saved Search for a Parent Object	Yes	Yes	Yes	Yes	Yes	Yes
Create Work Area for a Custom Top Level Object	Yes					
Add, Edit Custom or Standard fields to Local Search	Yes	No	No	No	No	No
Add, Edit Custom or Standard fields to Regional Search	Yes	No	No	No	No	No
Add or Edit Custom or Standard fields to Summary Table	Yes	Yes	Yes	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Creation Page	Yes	Yes	Yes	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Summary Form	Yes	Yes	Yes	Yes	Yes	Yes
Add Actions or Links to Summary Table	Yes	Yes	Yes	Yes	Yes	Yes
Add Actions or Links to Summary Form	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Picker	Yes	No	No	No	No	No

Create or Edit Regional Pane	Yes	No	No	No	No	No
Create or Edit Tree Nodes	Yes	No	No	No	No	No
Create or Edit Child or Related Object subtab	Yes	Yes	No	Yes	No	No
Create or Edit Context Link subtab	Yes	Yes	No	Yes	No	No
Create or Edit Common Component subtab	No	No	No	No	No	No
Create or Edit Web Content subtab	Yes	Yes	No	Yes	No	No
Create Mobile Pages	No	No	No	No	No	No
Create Tablet Pages	No	No	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom text attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Number attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	Yes	Yes	Yes	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	Yes	Yes	Yes	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	Yes	Yes	Yes	Yes	Yes	Yes
Find and get a record Standard or Custom object having custom attribute	Yes	Yes	Yes	Yes	Yes	Yes
Create an action script that triggers a Web service	No	No	No	No	No	No
Create or Edit or Save and Publish a Subject Area	Yes	Yes	Yes	No	Yes	Yes
Edit or Save republish a Subject Area	Yes	Yes	Yes	No	Yes	Yes
Create or Edit Object workflow	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete Field Updates	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete E-mail Notification	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete Outbound Message	Yes	Yes	Yes	Yes	Yes	Yes

Create or Edit or Delete Business-process Flow	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete Task Creation	Yes	Yes	Yes	No	No	No
Execute Object Workflow for all actions	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete E-mail Templates	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit a Relationship	Yes	Yes	Yes	Yes	Yes	Yes

Table (continued from above): Oracle Fusion Marketing objects

Customization Task Flow	Marketing Event Activity	Marketing Advertisement	Marketing Claim Version 2	Marketing Claim Settlement Version 2	Marketing Budget Entry Version 2	Marketing Fund Request Version 2	Marketing Response
Create or Edit Top Level Object							
Create or Edit Child Object for Standard and Custom Parent Object	No	No	Yes	No	No	Yes	Yes
Create or Edit Text Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Date-time Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Create or Edit Long-text Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Check-box Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Percentage Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Currency Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Fixed Choice List Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Dynamic Choice List Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Formula Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create Indexed Attribute	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with Script (using groovy)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with URL	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Links with URL	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Functions using groovy scripts	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit Saved Search for a Parent Object	No	No	Yes	No	Yes	Yes	Yes
Create Work Area for a Custom Top Level Object							
Add, Edit Custom or Standard fields to Local Search	No	No	No	No	No	No	No
Add, Edit Custom or Standard fields to Regional Search	No	No	No	No	No	No	No
Add or Edit Custom or Standard fields to Summary Table	No	Yes	Yes	No	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Creation Page	No	No	Yes	Yes	Yes	Yes	Yes

Add or Edit Custom or Standard fields to Summary Form	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Add Actions or Links to Summary Table	No	No	Yes	No	Yes	Yes	Yes
Add Actions or Links to Summary Form	No	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Picker	No	No	No	No	No	No	No
Create or Edit Regional Pane	No	No	No	No	No	No	No
Create or Edit Tree Nodes	No	No	No	No	No	No	No
Create or Edit Child or Related Object subtab	No	No	Yes	No	No	Yes	Yes
Create or Edit Context Link subtab	No	No	Yes	No	No	Yes	Yes
Create or Edit Common Component subtab	No	No	No	No	No	No	No
Create or Edit Web Content subtab	No	No	Yes	No	No	Yes	Yes
Create Mobile Pages	No	No	No	No	No	No	No
Create Tablet Pages	No	No	No	No	No	No	No

Create, Update and Delete a record in Standard or Custom object having custom text attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Number attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	No	No	Yes	Yes	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	No	No	Yes	Yes	Yes	Yes	Yes

Find and get a record Standard or Custom object having custom attribute	No	No	Yes	Yes	Yes	Yes	Yes
Create an action script that triggers a Web service	No	No	No	No	No	No	No
Create or Edit or Save and Publish a Subject Area	No	No	No	No	No	No	Yes
Edit or Save republish a Subject Area	No	No	No	No	No	No	Yes
Create or Edit Object workflow	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete Field Updates	Yes	Yes	Yes	Yes	No	Yes	Yes
Create, Edit or Delete E-mail Notification	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete Outbound Message	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete Business-process Flow	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete Task Creation	No	No	No	No	No	No	No

Execute Object Workflow for all actions	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit or Delete E-mail Templates	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Create or Edit a Relationship	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table (continued from above): Oracle Fusion Marketing objects

Customization Task Flow	Sales Lead Contacts	Sales Lead Resources	Sales Campaign
Create or Edit Top Level Object			
Create or Edit Child Object for Standard and Custom Parent Object	No	No	No
Create or Edit Text Attribute	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	Yes	Yes
Create or Edit Date-time Attribute	Yes	Yes	Yes
Create or Edit Long-text Attribute	Yes	Yes	Yes
Create or Edit Check-box Attribute	Yes	Yes	Yes
Create or Edit Percentage Attribute	Yes	Yes	Yes
Create or Edit Currency Attribute	Yes	Yes	Yes
Create or Edit Fixed Choice List Attribute	Yes	Yes	Yes
Create or Edit Dynamic Choice List Attribute	Yes	Yes	Yes
Create or Edit Formula Attribute	Yes	Yes	Yes
Create Indexed Attribute	Yes	Yes	Yes
Create or Edit Action with Script (using groovy)	Yes	Yes	Yes

Create or Edit Action with URL	Yes	Yes	Yes
Create or Edit Links with URL	Yes	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes
Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes	Yes	Yes
Create, Edit or Delete Object Functions using groovy scripts	Yes	Yes	Yes
Create or Edit Saved Search for a Parent Object	No	No	Yes
Create Work Area for a Custom Top Level Object			
Add, Edit Custom or Standard fields to Local Search	No	No	No
Add, Edit Custom or Standard fields to Regional Search	No	No	No
Add or Edit Custom or Standard fields to Summary Table	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Creation Page	No	No	Yes
Add or Edit Custom or Standard fields to Summary Form	No	No	Yes
Add Actions or Links to Summary Table	Yes	No	No
Add Actions or Links to Summary Form	No	No	No
Create or Edit Picker	No	No	No
Create or Edit Regional Pane	No	No	No
Create or Edit Tree Nodes	No	No	No
Create or Edit Child or Related Object subtab	No	No	No
Create or Edit Context Link subtab	No	No	No

Create or Edit Common Component subtab	No	No	No
Create or Edit Web Content subtab	No	No	No
Create Mobile Pages	No	No	No
Create Tablet Pages	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom text attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Number attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	Yes	Yes	No
Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	Yes	Yes	No

Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	Yes	Yes	No
Find and get a record Standard or Custom object having custom attribute	Yes	Yes	No
Create an action script that triggers a Web service	No	No	No
Create or Edit or Save and Publish a Subject Area	No	No	No
Edit or Save republish a Subject Area	No	No	No
Create or Edit Object workflow	No	Yes	No
Create or Edit or Delete Field Updates	No	Yes	No
Create, Edit or Delete E-mail Notification	No	Yes	No
Create or Edit or Delete Outbound Message	No	Yes	No
Create or Edit or Delete Business-process Flow	No	Yes	No
Create or Edit or Delete Task Creation	No	No	No
Execute Object Workflow for all actions	No	Yes	No
Create or Edit or Delete E-mail Templates	No	Yes	No
Create or Edit a Relationship	No	No	Yes

Oracle Fusion Sales

Using the Sales application container in the Application Composer, you can:

- Create custom objects
- Complete the tasks listed in the tables below for standard objects.

Customization Task Flow	Custom Object	Opportunity	Sales Competitor	Sales Reference Customer SDO	Opportunity Contact
Create or Edit Top Level Object	Yes				
Create or Edit Child Object for Standard and Custom Parent Object	Yes	Yes	Yes	Yes	No

Create or Edit Text Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Date-time Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Long-text Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Check-box Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Percentage Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Currency Attribute	Yes	Yes	No	No	No
Create or Edit Fixed Choice List Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Dynamic Choice List Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Formula Attribute	Yes	Yes	Yes	Yes	Yes
Create Indexed Attribute	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with Script (using groovy)	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with URL	Yes	Yes	Yes	Yes	Yes
Create or Edit Links with URL	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes

Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Functions using groovy scripts	Yes	Yes	Yes	Yes	Yes
Create or Edit Saved Search for a Parent Object	Yes	Yes	Yes	Yes	No
Create, Update or Delete roles for Custom Object	Yes				
Create, Update or Delete permissions for Custom Object	Yes				
View APM roles defined across the applications for Custom Object	Yes				
View or Edit policies defined for a role (Application Specific Custom Objects)	Yes				
Create Work Area for a Custom Top Level Object	Yes				
Add, Edit Custom or Standard fields to Local Search	Yes	No	No	No	No
Add, Edit Custom or Standard fields to Regional Search	Yes	No	No	No	No

Add or Edit Custom or Standard fields to Summary Table	Yes	Yes	Yes	Yes	No
Add or Edit Custom or Standard fields to Creation Page	Yes	Yes	Yes	Yes	No
Add or Edit Custom or Standard fields to Summary Form	Yes	Yes	Yes	Yes	No
Add Actions or Links to Summary Table	Yes	Yes	Yes	Yes	Yes
Add Actions or Links to Summary Form	Yes	Yes	Yes	Yes	No
Create or Edit Picker	Yes	No	No	No	No
Create or Edit Regional Pane	Yes	No	No	No	No
Create or Edit Tree Nodes	Yes	No	No	No	No
Create or Edit Child or Related Object subtab	Yes	Yes	Yes	Yes	No
Create or Edit Context Link subtab	Yes	Yes	Yes	Yes	No
Create or Edit Common Component subtab	Yes	No	No	No	No
Create or Edit Web Content subtab	Yes	Yes	Yes	Yes	No
Create Mobile Pages	Yes	Yes	Yes	Yes	No
Create Tablet Pages	Yes	Yes	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom text attribute	Yes	Yes	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Number attribute	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	Yes	Yes	Yes	Yes	Yes

Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	Yes	Yes	Yes	Yes	Yes
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	Yes	Yes	Yes	Yes	Yes
Find and get a record Standard or Custom object having custom attribute	Yes	Yes	Yes	Yes	Yes
Create an action script that triggers a Web service	No	No	No	No	No
Create, Edit or Save and Publish a Subject Area	Yes	Yes	Yes	No	No
Edit or Save republish a Subject Area	Yes	Yes	Yes	No	No
Create or Edit Object workflow	Yes	Yes	Yes	Yes	No
Create, Edit or Delete Field Updates	Yes	Yes	Yes	Yes	No
Create, Edit or Delete E-mail Notification	Yes	Yes	Yes	Yes	No
Create, Edit or Delete Outbound Message	Yes	Yes	Yes	Yes	No
Create, Edit or Delete Business-process Flow	Yes	Yes	Yes	Yes	No
Create, Edit or Delete Task Creation	Yes	Yes	Yes	Yes	No
Execute Object Workflow for all actions	Yes	Yes	Yes	Yes	No

Create, Edit or Delete E-mail Templates	Yes	Yes	Yes	Yes	No
Create or Edit a Relationship	Yes	Yes	Yes	Yes	No

Table (continued from above): Oracle Fusion Sales objects

Customizati Task Flow	Opportunity Revenue	Forecast Territory Details	Forecast Item	Partner	Programs	Program Objective Details	Program Benefits Details
Create or Edit Top Level Object							
Create or Edit Child Object for Standard and Custom Parent Object	No	No	No	Yes	Yes	No	No
Create or Edit Text Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Number Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Date Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Date-time Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Long-text Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Check-box Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Percentage Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Currency Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes

Create or Edit Fixed Choice List Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Dynamic Choice List Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Formula Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create Indexed Attribute	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with Script (using groovy)	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Action with URL	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Links with URL	Yes	No	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes	No	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes	No	Yes	Yes	Yes	Yes	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes	No	Yes	Yes	Yes	Yes	Yes

Create, Edit or Delete Object Functions using groovy scripts	Yes	No	Yes	Yes	Yes	Yes	Yes
Create or Edit Saved Search for a Parent Object	No	No	No	Yes	Yes	No	No
Create, Update or Delete roles for Custom Object							
Create, Update or Delete permissions for Custom Object							
View APM roles defined across the applications for Custom Object							
View or Edit policies defined for a role (Application Specific Custom Objects)							
Create Work Area for a Custom Top Level Object							
Add, Edit Custom or Standard fields to Local Search	No	No	No	Yes	Yes	No	No

Add, Edit Custom or Standard fields to Regional Search	No	No	No	No	No	No	No
Add or Edit Custom or Standard fields to Summary Table	Yes	No	Yes	Yes	Yes	Yes	Yes
Add or Edit Custom or Standard fields to Creation Page	No	No	No	Yes	Yes	No	No
Add or Edit Custom or Standard fields to Summary Form	No	No	No	Yes	No	No	No
Add Actions or Links to Summary Table	Yes	No	No	No	Yes	Yes	Yes
Add Actions or Links to Summary Form	No	No	No	No	Yes	No	No
Create or Edit Picker	No	No	No	No	No	No	No
Create or Edit Regional Pane	No	No	No	No	No	No	No
Create or Edit Tree Nodes	No	No	No	No	No	No	No
Create or Edit Child or Related Object subtab	No	No	No	Yes	Yes	No	No
Create or Edit Context Link subtab	No	No	No	No	No	No	No

Create or Edit Common Component subtab	No	No	No	No	No	No	No
Create or Edit Web Content subtab	No	No	No	No	No	No	No
Create Mobile Pages	No	No	No	No	No	No	No
Create Tablet Pages	No	No	No	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom text attribute	Yes		Yes	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom Number attribute	Yes		Yes	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	Yes		Yes	No	No	No	No

Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	Yes		Yes	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	Yes		Yes	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	Yes		Yes	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	Yes		Yes	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	Yes		Yes	No	No	No	No

Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	Yes		Yes	No	No	No	No
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	Yes		Yes	No	No	No	No
Find and get a record Standard or Custom object having custom attribute	Yes		Yes	No	Yes	No	No
Create an action script that triggers a Web service	No		Yes	No	No	No	No
Create, Edit or Save and Publish a Subject Area	No			No	Yes	No	No
Edit or Save republish a Subject Area	No			No	Yes	No	No
Create or Edit Object workflow	No	No	No	Yes	Yes	No	No
Create, Edit or Delete Field Updates	No	No	No	Yes	Yes	No	No
Create, Edit or Delete E-mail Notification	No	No	No	Yes	Yes	No	No

Create, Edit or Delete Outbound Message	No	No	No	Yes	Yes	No	No
Create, Edit or Delete Business-process Flow	No	No	No	Yes	Yes	No	No
Create, Edit or Delete Task Creation	No	No	No	No	No	No	No
Execute Object Workflow for all actions	No	No	No	Yes	Yes	No	No
Create, Edit or Delete E-mail Templates	No	No	No	Yes	Yes	No	No
Create or Edit a Relationship	No	No	No	Yes	Yes	No	No

Oracle Fusion Sales Catalog

Using the Sales Catalog application container in the Application Composer, you can complete the tasks listed in the table below.

Note

You cannot create custom objects using the Sales Catalog application container in the Application Composer.

Customization Task Flow	Product Group Details
Create or Edit Top Level Object	
Create or Edit Child Object for Standard and Custom Parent Object	No
Create or Edit Text Attribute	Yes
Create or Edit Number Attribute	Yes
Create or Edit Date Attribute	Yes
Create or Edit Date-time Attribute	Yes
Create or Edit Long-text Attribute	Yes
Create or Edit Check-box Attribute	Yes
Create or Edit Percentage Attribute	Yes
Create or Edit Currency Attribute	No
Create or Edit Fixed Choice List Attribute	Yes
Create or Edit Dynamic Choice List Attribute	No
Create or Edit Formula Attribute	Yes
Create Indexed Attribute	Yes

Create or Edit Action with Script (using groovy)	Yes
Create or Edit Action with URL	Yes
Create or Edit Links with URL	Yes
Create, Edit or Delete Object Rules (Validation Rules) using groovy scripts	Yes
Create, Edit or Delete Field Rules (Validation Rules) using groovy scripts	Yes
Create, Edit or Delete Triggers using groovy scripts	Yes
Create, Edit or Delete Object Functions using groovy scripts	
Create or Edit Saved Search for a Parent Object	No
Create Work Area for a Custom Top Level Object	
Add, Edit Custom or Standard fields to Local Search	No
Add, Edit Custom or Standard fields to Regional Search	No
Add or Edit Custom or Standard fields to Summary Table	No
Add or Edit Custom or Standard fields to Creation Page	No
Add or Edit Custom or Standard fields to Summary Form	Yes
Add Actions or Links to Summary Table	No
Add Actions or Links to Summary Form	No
Create or Edit Picker	No
Create or Edit Regional Pane	No
Create or Edit Tree Nodes	No
Create or Edit Child or Related Object subtab	No
Create or Edit Context Link subtab	No
Create or Edit Common Component subtab	No
Create or Edit Web Content subtab	No
Create Mobile Pages	No
Create Tablet Pages	No
Create, Update and Delete a record in Standard or Custom object having custom text attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Number attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Date attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Date-time attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Percentage attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Check-box attribute	No

Create, Update and Delete a record in Standard or Custom object having custom Fixed Choice List attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Dynamic Choice List attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Long Text attribute	No
Create, Update and Delete a record in Standard or Custom object having custom Currency attribute	No
Find and get a record Standard or Custom object having custom attribute	No
Create an action script that triggers a Web service	No
Create, Edit or Save and Publish a Subject Area	No
Edit, Save republish a Subject Area	No
Create or Edit Object workflow	No
Create, Edit or Delete Field Updates	No
Create, Edit or Delete E-mail Notification	No
Create, Edit or Delete Outbound Message	No
Create, Edit or Delete Business-process Flow	No
Create, Edit or Delete Task Creation	No
Execute Object Workflow for all actions	No
Create, Edit or Delete E-mail Templates	No
Create or Edit a Relationship	No

Oracle Fusion Common CRM Objects: Explained

You can customize a variety of pages and regions throughout Oracle Fusion CRM using the Oracle Fusion CRM Application Composer. The Application Composer lets you create custom fields and objects for a specific CRM application, which you then add for display at runtime. Objects can be standard or custom, and are attached to a specific application container, such as Oracle Fusion Marketing or Oracle Fusion Sales. Oracle Fusion Common CRM objects, however, are unique in that the customizations you make to these objects can be displayed in a variety of CRM applications, and in a variety of pages and regions, at runtime. This topic provides an overview of each Common CRM object that is part of the Trading Community Foundation. (This topic does not address the common component objects which are also listed within the Common CRM application container: the interaction, note, and task objects.)

The Trading Community Foundation Common CRM standard objects that you can extend are:

- Trading Community Address
- Trading Community Customer Contact Profile
- Trading Community Group Profile
- Trading Community Organization Profile

- Trading Community Person Profile
- Trading Community Relationship
- Trading Community Resource Profile

Note

Some of these objects are also documented in the related topic, Customizing Oracle Fusion Customer Center Pages: Explained.

Trading Community Address

The Trading Community Address object is an intersection object that stores a reference to a physical location/address, as well as a reference to a trading community organization, person, or group. Any extensions to this object are available for display in customer, contact, partner, and resource address user interfaces within Oracle Fusion CRM applications.

To display your extensions, such as custom fields, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the pages and regions listed below. You access the configuration pages in the Application Composer from the Pages node under the Trading Community Address object.

This table indicates, for the Trading Community Address object, which Application Composer configuration page lets you populate which page or region with your application customizations.

Configuration Page in Application Composer	Related Page or Region
Edit Address Creation Region	<ul style="list-style-type: none"> • Edit Customer - Address Create region in Customer Center • Edit Contact - Address Create region in Customer Center • Edit Partner Profile - Address Create region in Partner Center • Edit Partner - Partner Members - Address Create region within Partner Center
Edit Address Summary Table	<ul style="list-style-type: none"> • Edit Customer - Addresses region in Customer Center • Edit Contact - Addresses region in Customer Center • Edit Partner Profile - Address region in Partner Center • Edit Partner - Partner members - Address region within Partner Center
Edit Person Quick Creation Region	<ul style="list-style-type: none"> • Consumer Create region within Customer Center • Contact Create region within Customer Center • Person Create region within Customer Hub

Edit Organization Quick Creation Region	<ul style="list-style-type: none"> • Customer Create region within Customer Center • Organization Create region within Customer Hub
---	---

Note

Oracle Fusion CRM has provided multiple configuration capabilities for the Address object. Accordingly, do not use the Application Composer for these types of customizations:

- To define country-specific address formats, different labels for address elements, set address elements as required, and so on, use the Manage Address Formats task in the Setup and Maintenance work area.
 - To enable the list of values and validation for each country, use the Manage Geographies task in the Setup and Maintenance work area.
-

Trading Community Customer Contact Profile

The Trading Community Customer Contact Profile object stores all intrinsic information for all customer contact persons, in the context of the contact's relationship with a customer. For extensions specific to a customer contact relationship, use this object. By contrast, a standalone contact in Oracle Fusion CRM is a contact not tied to any customer and cannot be extended using this object. Instead, for extensions that apply to all person parties, use the Trading Community Person Profile object.

Any extensions to this object are available for display in various pages within Oracle Fusion CRM applications. Generic person extensions are also visible from the Oracle Fusion Customer Hub application in the Person work area.

To display your extensions, such as custom fields, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the pages and regions listed below. You access the configuration pages in the Application Composer from the Pages node under the Trading Community Customer Contact Profile object.

This table indicates, for the Trading Community Customer Contact Profile object, which Application Composer configuration page lets you populate which page or region with your application customizations.

Configuration Page in Application Composer	Related Page or Region
Edit Person Quick Creation Region	<ul style="list-style-type: none"> • Consumer Create region within Customer Center • Contact Create region within Customer Center
Edit Contact Summary Table	<ul style="list-style-type: none"> • Edit Customer - Contacts table in Customer Center • Edit Partner - Members table in Partner Center
Edit Contact Read-Only Region	<ul style="list-style-type: none"> • Edit Contact - Basic Information region in Customer Center

Edit Contact Details Region	<ul style="list-style-type: none"> • Edit Customer: Contact Details within Customer Center • Edit Organization: Relationships within Customer Hub • Edit Partner - Partner Members - Person Details region within Partner Center
-----------------------------	---

Trading Community Group Profile

The Trading Community Group Profile object stores all intrinsic information for all groups within the trading community of the deploying company. In Oracle Fusion CRM, groups could be customers of the deploying company. The Trading Community Group Profile object is a generic profile that is applicable to all types of groups in the system. The Sales Account object is applicable only to those groups that are sales accounts. The Group Profile object is a date effective entity, which means that every change or update that happens to any attribute of the object causes a new record to be created in the object with effective start and end dates. This maintains the profile history information for the object.

Note

Extend this object only when your customization is required for all groups within your trading community.

In Oracle Fusion CRM, groups are available as customers, so any extensions to this object are available for display in the Customer pages. Generic group extensions are also visible from the Oracle Fusion Customer Hub application in the Groups work area. Extensions are not available for display in Customer Center.

To display your extensions, such as custom fields, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the pages and regions listed below. You access the configuration pages in the Application Composer from the Pages node under the Trading Community Group Profile object.

This table indicates, for the Trading Community Group Profile object, which Application Composer configuration page lets you populate which page or region with your application customizations.

Configuration Page in Application Composer	Related Page or Region
Edit Summary Table	<ul style="list-style-type: none"> • Oracle Fusion Customer Hub application in the Groups work area
Edit Group Update Form	<ul style="list-style-type: none"> • Oracle Fusion Customer Hub application in the Groups work area
Edit Group Create Form	<ul style="list-style-type: none"> • Oracle Fusion Customer Hub application in the Groups work area

Trading Community Organization Profile

The Trading Community Organization Profile object stores all intrinsic information for all organizations within the trading community of the deploying company. Examples of organizations from a CRM perspective are customers (including sales prospects and external legal entities), competitors, and partners.

To specifically extend customer or partner objects, use the Sales Account object (under the Customer Center application container) or the Partner object (under the Sales application container). (The Trading Community Organization Profile object is a generic profile that is applicable to all types of organizations in the system. The Sales Account object is applicable only to those organizations that are sales accounts.) Since Prospect and Legal Entity customers don't have separate profiles, you can extend these customers only by using the Trading Community Organization Profile object. The Organization Profile object is a date effective entity, which means that every change or update that happens to any attribute of the object causes a new record to be created in the object with effective start and end dates. This maintains the profile history information for the object.

Note

Extend this object only when your customization is required for all organizations within your trading community.

Any extensions to this object are available for display in various pages within Oracle Fusion CRM applications. Generic organization extensions are also visible from the Oracle Fusion Customer Hub application in the Organization work area.

To display your extensions, such as custom fields, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the pages and regions listed below. You access the configuration pages in the Application Composer from the Pages node under the Trading Community Organization Profile object.

This table indicates, for the Trading Community Organization Profile object, which Application Composer configuration page lets you populate which page or region with your application customizations.

Configuration Page in Application Composer	Related Page or Region
Edit Organization Details Region	<ul style="list-style-type: none"> Customer Details region within Customer Center Organization Details region within Customer Hub Competitor Summary region within Sales
Edit Organization Quick Creation Region	<ul style="list-style-type: none"> Customer Create region within Customer Center Organization Create region within Customer Hub
Edit Organization Read-Only Region	<ul style="list-style-type: none"> Basic Information Region within Customer Center Basic Organization Information Region within Customer Hub

Trading Community Person Profile

The Trading Community Person Profile object stores all intrinsic information for all persons within the trading community of the deploying company. Examples of persons from a CRM perspective are customers (also called consumers),

contacts, internal employee resources, and partner members. To extend the customer contact relationship, use the Trading Community Customer Contact Profile object. To extend internal resources, use the Trading Community Resource Profile object. To extend person customers (also called consumers), use the Sales Account object (under the Customer Center application container). (The Trading Community Person Profile object is a generic profile that is applicable to all types of persons in the system. The Sales Account object is applicable only to those persons who are sales accounts.) The Person Profile object is a date effective entity, which means that every change or update that happens to any attribute of the object causes a new record to be created in the object with effective start and end dates. This maintains the profile history information for the object.

Note

Extend this object only when your customization is required for all persons within your trading community.

Any extensions to this object are available for display in various pages within Oracle Fusion CRM applications. Generic person extensions are also visible from the Oracle Fusion Customer Hub application in the Person work area.

To display your extensions, such as custom fields, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the pages and regions listed below. You access the configuration pages in the Application Composer from the Pages node under the Trading Community Person Profile object.

This table indicates, for the Trading Community Person Profile object, which Application Composer configuration page lets you populate which page or region with your application customizations.

Configuration Page in Application Composer	Related Page or Region
Edit Person Read-Only Region	<ul style="list-style-type: none"> Basic Information region within Customer Center (for Consumer or Person customer) Contact Basic Information within Customer Center Basic Person Information region within Customer Hub
Edit Person Details Region	<ul style="list-style-type: none"> Customer Details region within Customer Center (for Consumer or Person customer) Manage Contacts: Contact Details within Customer Center
Edit Person Quick Creation Region	<ul style="list-style-type: none"> Consumer Create region within Customer Center Contact Create region within Customer Center Person Create region within Customer Hub

Trading Community Relationship

The Trading Community Relationship object stores all intrinsic relationship information for all kinds of relationships between parties. Use this object to add attributes specific to a relationship between any two parties.

Note

Though a customer contact is also a type of relationship, use the Trading Community Customer Contact Profile object for customer contact relationship-specific extensions. Use this Trading Community Relationship object to extend only non-contact types of relationships.

Relationship Profile pages are regions that are visible in the Customer Hub. Accordingly, any extensions to this object are available for display in various pages within Oracle Fusion CRM applications. Generic person extensions are also visible from the Oracle Fusion Customer Hub application in the Person work area.

To display your extensions, such as custom fields, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the pages and regions listed below. You access the configuration pages in the Application Composer from the Pages node under the Trading Community Relationship object.

This table indicates, for the Trading Community Relationship object, which Application Composer configuration page lets you populate which page or region with your application customizations.

Configuration Page in Application Composer	Related Page or Region
Edit Relationship Summary Table	<ul style="list-style-type: none">• Edit Organization: Relationships within Customer Hub• Edit Person: Relationships within Customer Hub• Edit Group: Relationships within Customer Hub• Create Group: Group Members within Customer Hub
Edit Relationship Creation Region	<ul style="list-style-type: none">• Edit Organization: Relationships within Customer Hub• Edit Person: Relationships within Customer Hub• Edit Group: Relationships within Customer Hub• Create Group Members within Customer Hub

Trading Community Resource Profile

The Trading Community Resource Profile object defines users of the Oracle Fusion CRM application. They are either internal employee and contractor resources, or partner member resources. Use this object when specific extensions are required for persons in the system that are resources.

To display your extensions, such as custom fields, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the pages and regions listed

below. You access the configuration pages in the Application Composer from the Pages node under the Trading Community Resource object.

This table indicates, for the Trading Community Resource object, which Application Composer configuration page lets you populate which page or region with your application customizations.

Configuration Page in Application Composer	Related Page or Region
Edit Details Region	<ul style="list-style-type: none"> The Resource Profile details page is available in the Resource Directory work area, or from the Manage Resources task in the Setup and Maintenance work area.
Edit Search Region for the Workarea Summary Table	<ul style="list-style-type: none"> The search region is available in the Resource Directory work area, under Search Resources.

Customizing Oracle Fusion Customer Center Pages: Explained

You can customize a variety of pages and regions in Oracle Fusion Customer Center using the Oracle Fusion CRM Application Composer. The Application Composer lets you create custom fields and objects, which you then add for display in the runtime Customer Center application. To access the Application Composer, select **Application Composer** from the Navigator menu, under the Tools category.

Customizing Oracle Fusion Customer Center Pages Using the Application Composer

In general, every top-level CRM object has a work area, which includes an overview page, a creation page, and a details page. When you make changes to the object, those changes can be reflected in the object's associated work area. Customer Center, however, is unique in that its user interface pages do not include the traditional work area combination of overview page, creation page, and details page. Instead, Customer Center has a series of tree nodes that, when selected, display user interface pages, and even a single page can be associated with multiple business objects.

This table lists Customer Center pages, and the related objects that you can access in the Application Composer to customize those pages.

Customer Center Page	Customer Center Region	Application	Underlying Business Object
Consumer Profile	Addresses region	Common	Trading Community Address
Consumer Profile	Consumer Basic Information region	Common	Trading Community Person Profile
Consumer Profile	Consumer Details region	Common	Trading Community Person Profile
Consumer Profile	Sales Account region	Customer Center	Sales Account
Contact Profile	Address region	Common	Trading Community Address
Contact Profile	Basic Information region	Common	Trading Community Org Contact

Contact Profile	Contact Details region	Common	Trading Community Org Contact
Contact Profile	Contacts region (also known as the Contacts List)	Common	Trading Community Org Contact
Create Consumer page and also the Quick Create Consumer page	New fields are added at the bottom of the page	Customer Center	Sales Account
Create Consumer page and also the Quick Create Consumer page	No specific region	Common	Trading Community Person Profile
Create Contact page and also the Quick Create Contact page	No specific region	Common	Trading Community Org Contact
Create Customer page and also the Quick Create Customer page	Contact Information region	Common	Trading Community Org Contact
Create Customer page and also the Quick Create Customer page	Customer Information region	Common	Trading Community Org Profile
Create Customer page and also the Quick Create Customer page	New fields are added at the bottom of the page.	Customer Center	Sales Account
Customer Profile	Addresses region	Common	Trading Community Address
Customer Profile	Basic Information region	Common	Trading Community Org Profile
Customer Profile	Customer Details region	Common	Trading Community Org Profile
Customer Profile	Sales Account region	Customer Center	Sales Account
Edit Customer page	Team Members region	Customer Center	Sales Account Resource (child object to the Sales Account)
Overview > Summary tab	Sales Accounts region (also known as the Sales Account List)	Customer Center	Sales Account
Real-Time Search: Customers page	Search region, by way of Advanced Search > Add Fields	Customer Center	Sales Account

Note

To make only minor user interface changes to Customer Center pages without creating new objects or fields, use Oracle Composer instead of the Application Composer.

Oracle Fusion Common CRM Objects

The Oracle Fusion Common CRM objects that are associated with Customer Center pages are:

- Trading Community Org Profile

- Trading Community Person Profile
- Trading Community Address
- Trading Community Org Contact

This table indicates which Common objects populate which Customer Center pages and regions, as well as the Application Composer configuration pages where you can make user interface changes on those pages and regions.

Business Object	Configuration Page in Application Composer	Related Customer Center Page	Related Customer Center Region
Trading Community Org Profile	Edit Customer Quick Creation Form	Create Customer page and also the Quick Create Customer page	Customer Information region
Trading Community Org Profile	Edit Read Only Form	Customer Profile	Basic Information region
Trading Community Org Profile	Edit Details Form	Customer Profile	Customer Details region
Trading Community Person Profile	Edit Contact/Consumer Quick Creation Form	Create Consumer page and also the Quick Create Consumer page	No specific region
Trading Community Person Profile	Edit Read Only Form	Consumer Profile	Consumer Basic Information region
Trading Community Person Profile	Edit Details Form	Consumer Profile	Consumer Details region
Trading Community Address	Edit Detail Form	Customer Profile	Addresses region
Trading Community Address	Edit Detail Form	Contact Profile	Addresses region
Trading Community Address	Edit Detail Form	Consumer Profile	Addresses region
Trading Community Org Contact	Edit Creation Form	Create Customer page and also the Quick Create Customer page	Contact Information region
Trading Community Org Contact	Edit Creation Form	Create Contact page and also the Quick Create Contact page	No specific region
Trading Community Org Contact	Edit Read Only Form	Contact Profile	Basic Information region
Trading Community Org Contact	Edit Contact Details Form	Contact Profile	Contact Details region
Trading Community Org Contact	Edit Summary Table	Contact Profile	Contacts region (also known as the Contacts List)

Oracle Fusion Customer Center Objects

The Oracle Fusion Customer Center objects that are associated with Customer Center pages are:

- Sales Account

- Sales Account Resource (child of the Sales Account)

This table indicates which Customer Center objects populate which Customer Center pages and regions, as well as the Application Composer configuration pages where you can make user interface changes on those pages and regions.

Business Object	Configuration Page in Application Composer	Related Customer Center Page	Related Customer Center Region
Sales Account	Edit Creation Form	Create Customer page and also the Quick Create Customer page	New fields are added at the bottom of the page.
Sales Account	Edit Creation Form	Create Consumer page and also the Quick Create Consumer page	New fields are added at the bottom of the page.
Sales Account	Edit Details Form	Customer Profile	Sales Account region
Sales Account	Edit Details Form	Consumer Profile	Sales Account region
Sales Account	Edit Summary Table	Overview, then Summary tab	Sales Accounts region (also known as the Sales Account List)
Sales Account	Not applicable. Custom fields are automatically available from the list of additional fields.	Real-Time Search: Customers page	Search region, by way of Advanced Search, then Add Fields
Sales Account	Edit Regional Panes	Customer Work Area	Panes in the regional area
Sales Account Resource (child object to the Sales Account)	Edit Summary Table	Edit Customer: Sales Account Team page	Team Members region

Using the Pages Overview Page

To add custom fields to the Customer Center pages listed in the tables above, use the Application Composer's various configuration pages. You access the configuration pages in the Application Composer from each object's Pages Overview page. Before you access the configuration pages, you must have already created your custom fields using the Application Composer.

To access the Pages Overview page:

1. Select either the **Common** or **Customer Center** application on the main Overview page.
2. In the object tree, select the object you want to customize.
3. Select the **Pages** node.
4. On the Pages Overview page, select the configuration page hyperlink related to the Customer Center page that you want to customize.

Adding Custom Reports Using Oracle Composer

The customizations that you can make in Customer Center also include the creation of reports. This type of customization does not involve the creation of new fields or objects within the Application Composer. Instead, create a report using BI Answers, save the report to the Resource Catalog, and then use

Oracle Composer to add the report to two pages: the Customer Snapshot and the Customer Overview, Analysis tab.

Note

The reports that you add to the Customer Snapshot are within the context of a single customer, because you view a single customer when viewing the Snapshot. The reports that you add to the Analysis tab on the Customer Overview provide context across multiple customers, because you are viewing multiple customers on the Analysis tab.

Customizing Oracle Fusion Sales Forecasting Pages: Explained

You can customize a variety of regions in Oracle Fusion Sales Forecasting using the Oracle Fusion CRM Application Composer. The Application Composer lets you create custom fields and objects, which you then add for display in the runtime Sales Forecasting application. To access the Application Composer, select **Application Composer** from the Navigator menu, under the Tools category.

Understanding Which Oracle Fusion Sales Forecasting Pages Are Extensible

To customize Sales Forecasting pages, you need to know which pages and regions are extensible, and which objects to select in the Application Composer to customize those pages.

The Oracle Fusion Sales objects that are associated with Sales Forecasting pages are:

- Forecast Item
- Forecast Territory Details

This table lists Sales Forecasting pages and regions, and the related objects that you can access in the Application Composer to customize those pages. For example, the Forecast Items tab on the Edit Forecast page is extensible. To create custom fields that you can later add to the Forecast Items tab, you must select the Forecast Item object in the Application Composer and create your custom fields.

Sales Forecasting Page	Sales Forecasting Region	Application	Underlying Business Object
Edit Forecast page	Territory Details region (expand the Show More option)	Sales	Forecast Territory Details
Edit Forecast page	Forecast Items tab	Sales	Forecast Item (child of Forecast Territory Details)

Next, expose those custom fields on the Forecast Items tab by accessing the appropriate Application Composer configuration page, listed in the following section.

Note

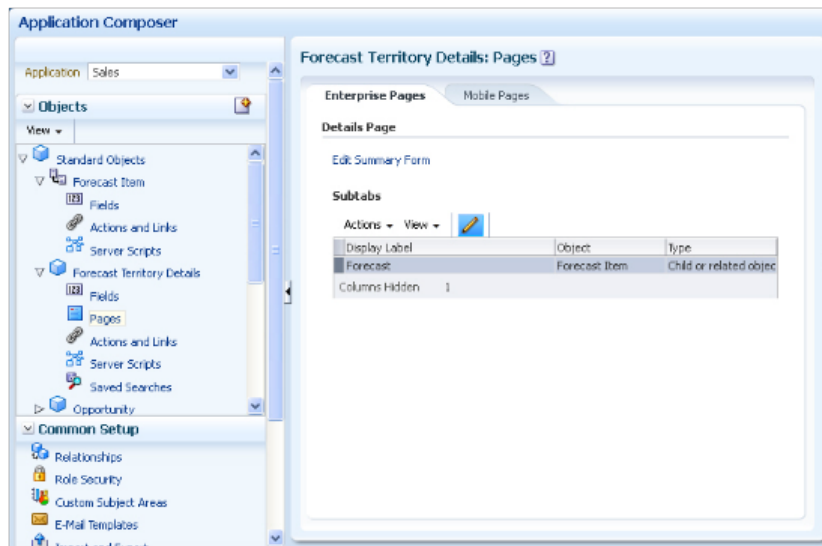
To make only minor user interface changes to Sales Forecasting pages without creating new objects or fields, use Oracle Composer instead of the Application Composer.

Adding Your Changes to the Runtime Application

To add custom fields to the Sales Forecasting regions listed in the table above, first create your custom fields using the Application Composer. Next, use the Application Composer's configuration pages to add those custom fields to the desired Sales Forecasting regions. You access the configuration pages in the Application Composer from the Pages node under the Forecast Territory Details object.

To access the Pages Overview page:

1. Select the **Sales** application on the main Overview page.
2. In the object tree, select the Forecast Territory Details object.
3. Select the **Pages** node.



4. On the Pages Overview page, select the configuration page hyperlink related to the Sales Forecasting region that you want to customize.

This table indicates which Sales objects populate which Sales Forecasting pages and regions, as well as the Application Composer configuration pages where you can make user interface changes on those pages and regions.

Business Object	Configuration Page in Application Composer	Related Sales Forecasting Page	Related Sales Forecasting Region
Forecast Item (child of Forecast Territory Details)	Select the Pages node for the Forecast Territory Details object, select the Forecast subtab row, and click the Edit icon to navigate to the Edit Subtab: Child or Related Object page.	Edit Forecast page	Forecast Items tab

Forecast Territory Details	Select the Pages node for this object, then click Edit Summary Form.	Edit Forecast page	Territory Details region <hr/> Note To view your custom fields in the Territory Details region, you must expand the Show More option which appears above the Forecast Items tab. <hr/>
----------------------------	--	--------------------	--

Application Composer: Creating Object Workflows

Object Workflows: Explained

The object workflows represent orchestrated business processes that are invoked based on the conditions that you define for triggering such processes. You must have administrative privileges to configure object workflows.

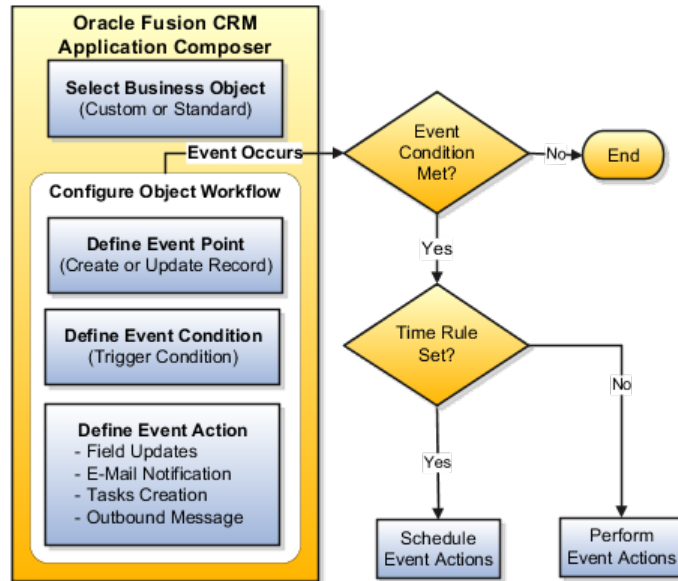
To configure an object workflow, select an object for which you want to create a workflow and specify the event point, event condition, and event action for that object.

This figure illustrates the create object workflow page.

The screenshot shows the 'Create Object Workflow' dialog box. It has a title bar with a question mark icon and 'Save' and 'Cancel' buttons. The main area is divided into several sections:

- Object:** A dropdown menu showing 'Opportunity'.
- Name:** A text input field containing 'notify team'.
- Description:** A large text area.
- Active:** A checkbox that is checked.
- Event Point and Condition:**
 - Event Point:** Two radio buttons: 'When a record is created' (selected) and 'When a record is updated'.
 - Condition:** A text area with two examples:
 - Example 1: `Status == 'IN_PROGRESS'`
 - Example 2: `if (isAttributeChanged('Status') && Status == 'IN_PROGRESS') return true;`
- Actions:** A list of actions with checkboxes and icons:
 - ☒ Field Updates
 - ☒ E-Mail Notification
 - ☒ Task Creation
 - ☒ Outbound Message

This figure illustrates the overall process for configuring and triggering object workflows.



Following are some of the common aspects of object workflow definition:

- **Business Object:** A business object or an object can be a standard object delivered with the product or a custom object that you define based on your business need.
- **Event Point:** An event point is associated with an object and is an instance when an event occurs. Event points are of the following types:
 - When a record is created.
 - When a record is updated.
- **Event Condition:** An event condition is a trigger for invoking object workflows. It is a combination of event point and an expression that supports logical, math operations, or field-value lookups. Defining an expression prevents the triggering of the object workflow each time an update or create event point occurs.
- **Event Action:** An event action determines what action is expected from an object workflow when the conditions set for an object are met.

Event actions are of the following types:

- Field Updates.
- E-Mail Notification.
- Task Creation.
- Outbound Message.

Oracle Fusion CRM Application Composer

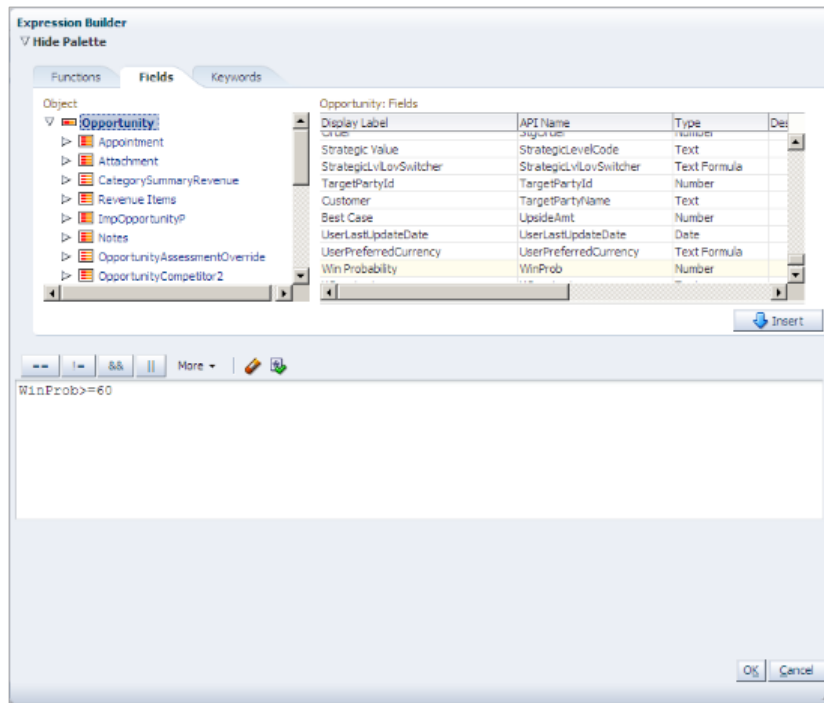
You can use the application composer to customize standard business objects delivered as is or create custom business objects based on business needs. You can use both **Standard Objects** and **Custom Objects** to define object workflows.

How you create and configure these objects determines and enforces that only the data relevant to the object in context is available for selection when defining a workflow. For example, when you define a trigger condition, an expression editor lists only those fields that are relevant to the object that you have selected.

Expression Editor

An expression editor supports building logical and math operations, including field lookups that you can optionally use to define trigger condition. Fields in the expression editor are populated based on the object for which you are defining a workflow. The expression editor displays a warning if your expression contains an invalid attribute name. However, you must confirm whether the attribute name is actually invalid. If an attribute exists which was created at runtime, then you can safely ignore the warning.

This figure illustrates the expression editor using which you can define groovy conditions for object workflows.



Use the expression editor to write Groovy-based application logic that determines when an object workflow is triggered. Some examples of the raise conditions you can use include the following:

Example 1:

```
Status=='IN_PROGRESS' && BudgetAvailableDate==Today() + 30
```

Example 2:

```
if (isAttributeChanged('PrimaryContactPartyName') &&
PrimaryContactPartyName =='Business World') return true; else return
false;
```

Example 3:

```
WinProb>10 || WinProb<50
```

In addition, keep these points in mind when using the expression editor to build raise conditions:

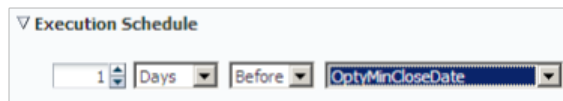
- Ensure that you return a valid boolean as part of your raise condition. Returning a non-boolean value could lead to runtime errors.
- Use `return true` or `return false` to explicitly return the boolean value and code indentation when the evaluation logic is complex, to minimize risk of runtime errors.
- For lookup values, use the lookup code instead of the display value.
- Use `<`, `>`, `==` for comparison.
- For the event point **When a record is updated**, avoid redundant calls of the actions by always specifying which field change should trigger the object workflow, using the function `isAttributeChanged`.
- Be aware of Groovy-type coercion, if you are not returning an explicit boolean value.
- Use the logging capability to debug your condition and review the generated log by selecting **Runtime Messages** in the Common Setup pane.

For more information, see the Developer's Guide to Scripting in Oracle Fusion CRM Application Composer on My Oracle Support at <https://support.oracle.com>.

Execution Schedule

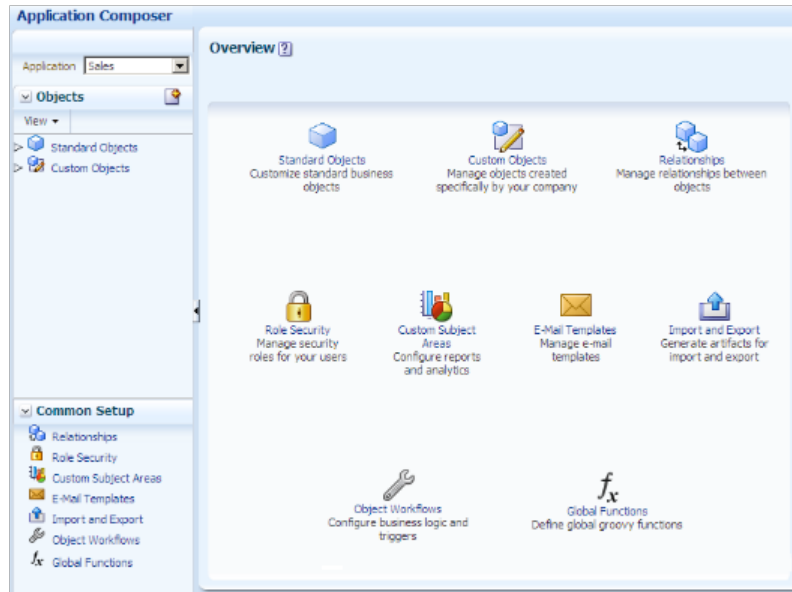
While defining an event action, you can optionally set an execution schedule that governs when an event action should happen. Else, the event actions are run immediately. When multiple event actions are configured, the Field Updates happen first before any other event action is triggered. For event actions other than Field Updates, there is no particular sequence and the event actions are triggered based on whether or how you configure the execution schedule. You can set a time rule for an event action based on whether that event action must occur after or before the triggering of a workflow or the occurrence of a date, and also specify the time duration in **Hours**, **Days**, or **Weeks**. If you schedule an event action for a time in the past, the event action is executed immediately after it is triggered.

This figure illustrates the execution schedule region.

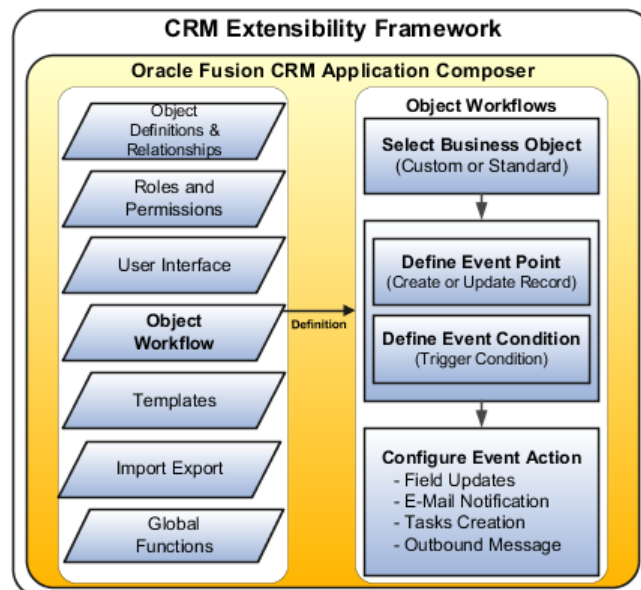


Object Workflows and Oracle Fusion CRM Application Composer: How They Work Together

Oracle Fusion CRM Application Composer is a browser-based configuration tool. It is part of the CRM Extensibility Framework that provides a mechanism to extend applications. You must have administrative privileges to work with and configure object workflows using the application composer.



This figure illustrates how the Oracle Fusion CRM Application Composer is used for defining object workflows.



Oracle Fusion CRM Application Composer

You can use the application composer to customize standard business objects delivered as is or create custom business objects based on business needs. You can use both **Standard Objects** and **Custom Objects** to define object workflows. How you create and configure these objects determines and enforces that only the data relevant to the object in context is available for selection when defining a workflow. For example, when you define a trigger condition, an expression editor lists only those fields that are relevant to the object that you have selected.

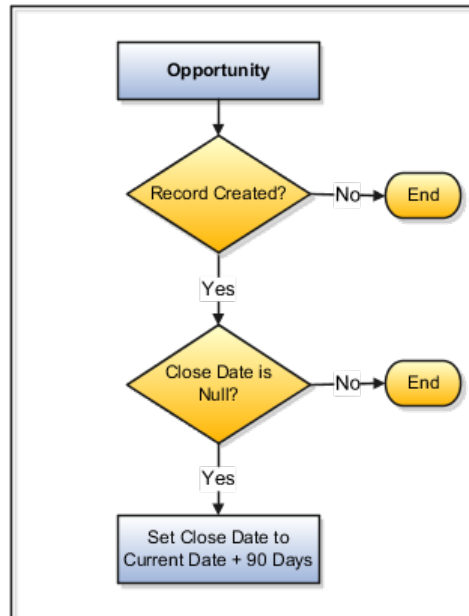
Object Workflows: Examples

These examples illustrate the business scenarios where object workflows can be employed to automate business processes.

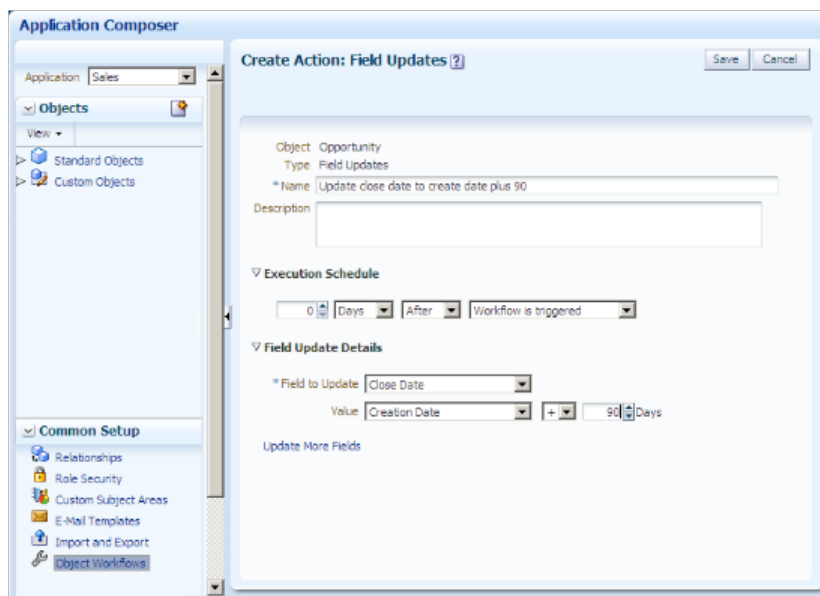
Scenario

In a sales division, the management plans to set an automated business process where each new opportunity must have an initial close date set automatically to 90 days from the day of creation of an opportunity. Being an administrator, you must create an object workflow based on management directions.

This figure illustrates an example of an event action triggered as part of object workflow definition.



This figure illustrates the create field updates page that you invoke from object workflow creation page after defining the trigger condition.



To create a workflow, select **Opportunity** as an object and set the event point as **When a record is created**. Use the expression editor to set the event condition as **Close Date** is **Null**.

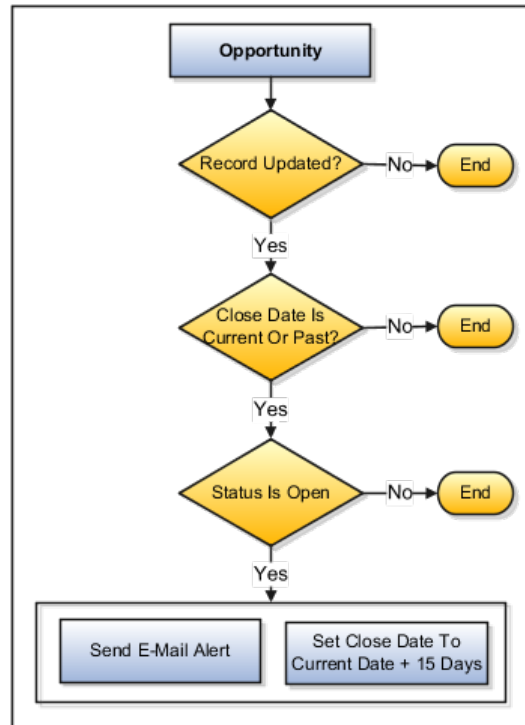
After defining the trigger condition, select the event action as **Field Updates** and use the expression editor to set the **Close Date** to **Current Date** plus 90 days;

hence, when a user creates a new **Opportunity** in the system, the workflow is automatically activated setting the **Close Date** to 90 days from the creation date.

Scenario

In a sales division, the management plans to set a business process that when an opportunity is updated whose close date is current or past and the status is open, a reminder e-mail should be sent to specified recipients and the close date should be extended by 15 days. Being an administrator, you must create an object workflow based on management directions.

This figure illustrates an example of multiple event actions triggered as part of object workflow definition.



This figure illustrates the create e-mail notification page that you invoke from object workflow creation page after defining the trigger condition.

The screenshot shows the 'Create Action: E-Mail Notification' dialog box. The 'Object' is 'Opportunity' and the 'Type' is 'E-Mail Notification'. The 'Name' is 'Notify Opportunity Team'. The 'Description' field is empty. Under 'Execution Schedule', the 'E-Mail Details' section is expanded. The 'E-Mail Template' is 'Oppty_Template_Team'. The 'Recipient Type' is 'Relative users on record'. The 'Available Recipients' list includes 'Record creator', 'Last record updater', and several 'Opportunity Team Member' roles. The 'Selected Recipients' list includes 'Opportunity Team Member'.

To create an object workflow with two event actions, select **Opportunity** object and set the event point as **When a record is updated**. Use the expression editor

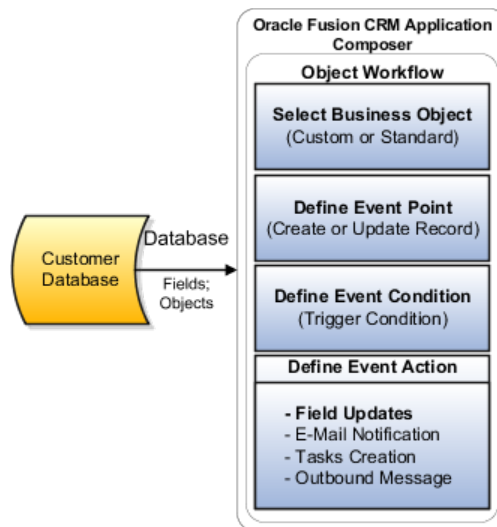
to set an event condition as **Close Date** is less than or equal to **Current Date** and **Status** is **Open**.

After defining the trigger condition, select an event action as **Field Updates** and use the expression editor to define an expression to set the **Close Date** to an additional 15 days. Create another event action as **E-Mail Notification** and specify the **Recipients**; hence, when an **Opportunity** is updated, the workflow checks the event conditions. If the conditions are met, an e-mail is sent to the specified recipients and the **Close Date** is extended by 15 days.

Object Workflows and Field Updates: How They Work Together

As part of object workflow definition, you can configure automatic field updates by specifying what fields you want to update and the value to apply to those fields. You must have administrative privileges to configure object workflows for field updates.

This figure illustrates how the field updates are configured and invoked.



To define an object workflow for Field Updates, select an **Object** and provide a meaningful **Name** for the workflow. Define the event trigger conditions and select **Field Updates** as the event action to configure and update one or more fields for the selected object.

For configuring Field Updates event action, define the **Name** and **Description** for the field update action and optionally set the **Execution Schedule** for triggering such updates. When you select a field to update, the options on the page dynamically change for configuring field update actions depending on the field type. For example, if you select a field of type **Date**, a **Date Picker** appears.

In case of static choice list, the list of field values can either be in an alphabetical order or in the display sequence such as **High**, **Medium**, or **Low**, as defined. For updating static fields that have a display sequence, you can specify whether the field is to be populated with next value in list or populated with previous value in list during field updates. For example, assuming a display sequence of **High**, **Medium**, or **Low**, you can change **High** to **Medium** (populate with next value) or **Low** to **Medium** (populate with previous value) during field updates. However, if the current value is **Low**, and the event action has been configured using **Populate with next value in the list**, the field is not updated when the action is executed, because **Low** is already the last value in the sequence. Use **Update More Fields** option to select additional fields for an update.

Same-Object Field Updates

The field-updates event action acts on the same object that raises the event. For example, when you create an **Opportunity** object and do not specify the **Close Date**, you can define a Field Update action that sets the **Close Date** automatically to 90 days after the creation date.

Execution Schedule

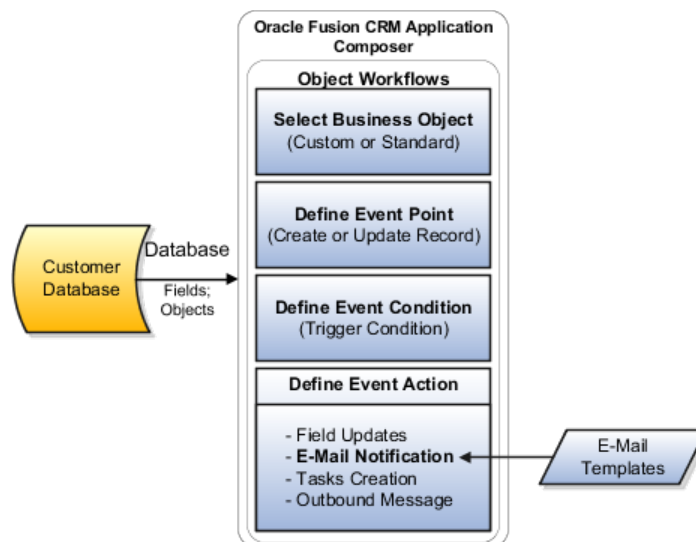
While defining an event action, you can optionally set an execution schedule that governs when an event action should happen. Else, the event actions are run immediately. When multiple event actions are configured, the Field Updates happen first before any other event action is triggered. For event actions other than Field Updates, there is no particular sequence and the event actions are triggered based on whether or how you configure the execution schedule. You can set a time rule for an event action based on whether that event action must occur after or before the triggering of a workflow or the occurrence of a date, and also specify the time duration in **Hours**, **Days**, or **Weeks**. If you schedule an event action for a time in the past, the event action is executed immediately after it is triggered.

Object Workflows and E-Mail Notification: How They Work Together

You can use object workflows to send e-mail communications to the intended recipients as an automated process. You must have administrative privileges to configure object workflows for e-mail notifications.

For example, you can set up a workflow that when the win probability is updated for an opportunity, an e-mail alert is automatically sent out to the resources who are assigned that opportunity along with the required information.

This figure illustrates how e-mail notifications are configured and invoked as part of object workflows.



To define an object workflow for E-Mail Notification, select an **Object** and provide a meaningful **Name** and **Description** for the workflow. Define the event trigger conditions and select **E-Mail Notification** as the event action to configure one or more notifications for the selected object.

For configuring E-Mail Notification as an event action, provide the **Name** and **Description** for the e-mail notification action and optionally set the **Execution Schedule** for triggering such notifications. Search an e-mail template by using

one or more **Search** parameters such as full or partial **Name**, **Description**, or **Subject**, or by specifying whether the template is **Active**. Select a template to use and provide the information that you require in the e-mail alert. The fields that you select for an e-mail alert are populated during runtime.

Recipient Types

Use the **Recipient Types** list box to select and send e-mail notifications to a list of resources or enter **Specific e-mail addresses** separated by a comma. Depending on the **Recipient Type**, you can configure e-mail notifications for relative users, roles, or specific users. A user can be, for example, creator of the object for which event action is defined, the person who last updated the record, direct or indirect subordinates, or sales resources working as a team, and so on.

Execution Schedule

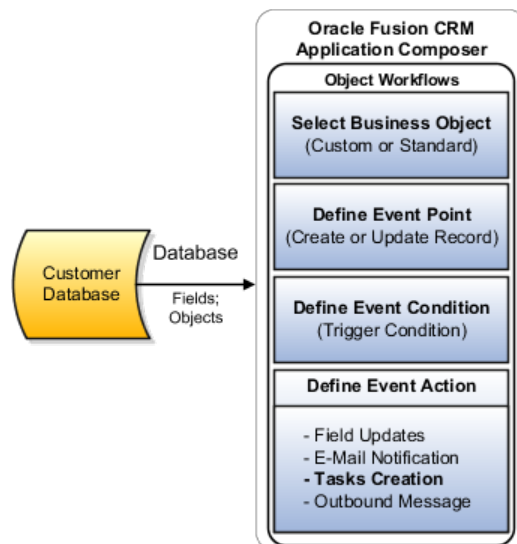
While defining an event action, you can optionally set an execution schedule that governs when an event action should happen. Else, the event actions are run immediately. When multiple event actions are configured, the Field Updates happen first before any other event action is triggered. For event actions other than Field Updates, there is no particular sequence and the event actions are triggered based on whether or how you configure the execution schedule. You can set a time rule for an event action based on whether that event action must occur after or before the triggering of a workflow or the occurrence of a date, and also specify the time duration in **Hours**, **Days**, or **Weeks**. If you schedule an event action for a time in the past, the event action is executed immediately after it is triggered.

Object Workflows and Task Creation: How They Work Together

Oracle Fusion tasks integrate with BPEL human tasks and are available in the BPEL worklist application. You must have administrative privileges to configure object workflows for Oracle Fusion tasks.

You can configure object workflows to trigger auto-creation and assignment of tasks for an object. You can define tasks event action for only those objects that are configured to support the creation of tasks. For other objects, the tasks option is disabled.

This figure illustrates how tasks are configured and invoked as part of object workflows.



To define an object workflow with a Task Creation event action, select an **Object** and provide a meaningful **Name** and **Description** for the workflow. Define the event trigger conditions and select **Task Creation** as the event action to configure a task for the selected object.

For configuring Task Creation event action, define the **Name** and **Description** for the task action and optionally set the **Execution Schedule**.

If an object for which you are defining a Task Creation event action contains a customer, that customer is automatically included in the created task.

Use the **Task Details** to configure a task based on your requirements.

- Select a **Subject** and **Description**, as appropriate, for your task notification.
- Select the **Start Date** and **Due Date** for a task. When you select a date, you can also provide logical conditions for these dates, for example, a **Due Date** must be 30 days after the **Start Date**.
- Select the **Owner** and **Assignees** for a task. The values in these lists change depending on the object for which the task is being defined. For example, for an opportunity object, the **Assignees** list would include the owner, resource team, resource team with different access levels, member functions, and so on.
- Selecting the **Category** to which a task belongs, such as meeting, administration, and so on.
- Assigning a **Priority** to the task. The default priority is 3.

For example, when an opportunity reaches a **Close Date** and the status is still **Open**, you can use the Task Creation event action to assign a follow up task to the owner of the opportunity.

Execution Schedule

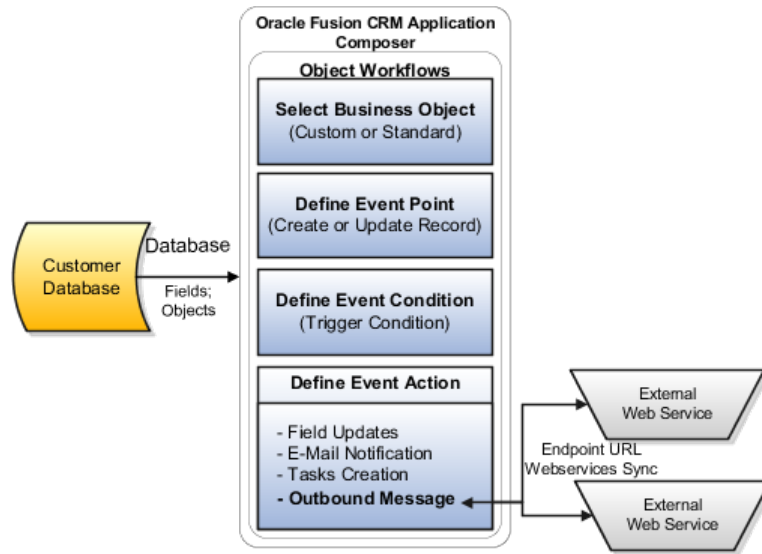
While defining an event action, you can optionally set an execution schedule that governs when an event action should happen. Else, the event actions are run immediately. When multiple event actions are configured, the Field Updates happen first before any other event action is triggered. For event actions other than Field Updates, there is no particular sequence and the event actions are triggered based on whether or how you configure the execution schedule. You can set a time rule for an event action based on whether that event action must occur after or before the triggering of a workflow or the occurrence of a date, and also specify the time duration in **Hours**, **Days**, or **Weeks**. If you schedule an event action for a time in the past, the event action is executed immediately after it is triggered.

Object Workflows and Outbound Message: How They Work Together

You can configure an object workflow to send an outbound message to a Web service at a specified endpoint URL. An endpoint URL is an external Web service that shares data with Oracle Fusion CRM.

You must have administrative privileges to configure object workflows.

This figure illustrates how outbound messages are invoked as part of object workflows.



To define an object workflow for outbound messages, select an **Object** and provide a meaningful **Name** and **Description** for the workflow. Define the event trigger conditions and select **Outbound Message** as the event action to send outbound messages for the selected object.

For configuring an outbound message event action, define the **Name** and **Description** for the outbound message action and set the **Execution Schedule** for triggering the action. Provide the **endpoint URL** of the external Web service, for example, an endpoint URL can be `http://GlobalFusion:7011/OMTestOpportunity/OutboundMessageServiceSoapHttpPort`. At runtime, a service data object containing details of the object on which the object workflow is defined is sent to the specified endpoint URL. For example, you can define an object workflow that automatically sends, based on a trigger, an outbound message containing an object like opportunity, lead, or account details from one system to another.

Outbound messages are queued up before they are processed. You can monitor the outbound messages including the history and the queued up messages and their statuses using the Oracle Enterprise Manager.

For more information about Oracle Enterprise Manager, see the Oracle Fusion Middleware Administrator's Guide.

As part of object workflow definition, if a field update event action is also scheduled along with outbound messages event action, the field updates event action is triggered first so that the outbound messages contains the updated data.

Execution Schedule

While defining an event action, you can optionally set an execution schedule that governs when an event action should happen. Else, the event actions are run immediately. When multiple event actions are configured, the Field Updates happen first before any other event action is triggered. For event actions other than Field Updates, there is no particular sequence and the event actions are triggered based on whether or how you configure the execution schedule. You can set a time rule for an event action based on whether that event action must occur after or before the triggering of a workflow or the occurrence of a date, and also specify the time duration in **Hours**, **Days**, or **Weeks**. If you schedule an

event action for a time in the past, the event action is executed immediately after it is triggered.

Creating Object-Specific Web Services: Explained

You can configure an object workflow to send an outbound message to a third-party Web service at a specified endpoint URL. An endpoint URL is an external Web service that receives data with Oracle Fusion Customer Relationship Management (CRM) applications. The third-party Web service must conform to the service WSDL defined by Oracle Fusion CRM.

This topic provides the following:

- How to configure a Web service
- An example of the OutboundMessageService.wsdl
- An example of the OutboundMessageService.xsd
- The security policies available

Configuring a Web Service

To configure a Web service, you must replace the parameters in the OutboundMessageService.xsd file using the Oracle Fusion Web service instructions.

Navigate to Oracle Enterprise Repository (OER):

1. For a standard object, search for **ADF Service** in OER by object name.
For custom objects, search for the generic Web service for all custom objects in the corresponding application, for example, Sales Custom Business Object.
2. Extract the .xsd files from the live environment URL using the information provided under the service.
3. Replace the parameters in OutboundMessageService.xsd with the names for the object of your interest.

The parameters are marked in the .xsd file as parameters, \$OBJECT_TARGET_NAMESPACE\$ and \$OBJECT_NAME\$.

For more information about Oracle Enterprise Repository, see Oracle Fusion Middleware Configuration Guide for Oracle Enterprise Repository.

Examples of OutboundMessageService.wsdl and OutboundMessageService.xsd are provided here for reference.

WSDL File Example

This section includes an example of the OutboundMessageService.wsdl, for your reference.

```
//Sample OutboundMessageService.wsdl
<wsdl:definitions
  name="OutboundMessageService"
  targetNamespace="http://xmlns.oracle.com/apps/crmCommon/content/
outboundMessage/"
  xmlns:errors="http://xmlns.oracle.com/adf/svc/errors/"
```

```

    xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
    xmlns:tns="http://xmlns.oracle.com/apps/crmCommon/content/
outboundMessage/"
    xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
    xmlns:types="http://xmlns.oracle.com/apps/crmCommon/content/
outboundMessage/types/"
  >
    <wSDL:import namespace="http://xmlns.oracle.com/adf/svc/errors/"
location="ServiceException.wSDL"/>
    <wSDL:types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
    <import namespace="http://xmlns.oracle.com/apps/crmCommon/content/
outboundMessage/types/"
schemaLocation="OutboundMessageService.xsd"/>
    </schema>
    </wSDL:types>
    <wSDL:message name="OutboundMessageService_processOutboundMessage">
    <wSDL:part name="parameters" element="types:processOutboundMessage"/>
    </wSDL:message>
    <wSDL:message
name="OutboundMessageService_processOutboundMessageResponse">
    <wSDL:part name="parameters"
element="types:processOutboundMessageResponse"/>
    </wSDL:message>
    <wSDL:portType name="OutboundMessageService">
    <wSDL:documentation/>
    <wSDL:operation name="processOutboundMessage">
    <wSDL:input message="tns:OutboundMessageService_processOutboundMessage"/>
    <wSDL:output
message="tns:OutboundMessageService_processOutboundMessageResponse"/>
    <wSDL:fault name="ServiceException" message="errors:ServiceException"/>
    </wSDL:operation>
    </wSDL:portType>
    <wSDL:binding name="OutboundMessageServiceSoapHttp"
type="tns:OutboundMessageService">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/
soap/http"/>
    <wSDL:operation name="processOutboundMessage">
    <soap:operation soapAction="http://xmlns.oracle.com/apps/crmCommon/
content/outboundMessage/processOutboundMessage"/>
    <wSDL:input>
    <soap:body use="literal"/>
    </wSDL:input>
    <wSDL:output>
    <soap:body use="literal"/>
    </wSDL:output>
    <wSDL:fault name="ServiceException">
    <soap:fault name="ServiceException" use="literal" encodingStyle=""/>
    </wSDL:fault>
    </wSDL:operation>
    </wSDL:binding>
    <wSDL:service name="OutboundMessageService">
    <wSDL:port name="OutboundMessageServiceSoapHttpPort"
binding="tns:OutboundMessageServiceSoapHttp">
    <soap:address location="http://adc2111013:7101/OMInterface/
OutboundMessageService"/>
    </wSDL:port>
    </wSDL:service>
  </wSDL:definitions>

```

XSD File Example

This section provides an example of the OutboundMessageService.xsd, for your reference. The parameters are marked in the .xsd file as parameters, \$OBJECT_TARGET_NAMESPACE\$ and \$OBJECT_NAME\$.

//Sample: OutboundMessageService.xsd

```
<schema elementFormDefault="qualified" targetNamespace="http://
xmlns.oracle.com/apps/crmCommon/content/outboundMessage/types/"
  xmlns:ns0="http://xmlns.oracle.com/adf/svc/errors/"
  xmlns:ns1="$OBJECT_TARGET_NAMESPACE$"
  xmlns:ns2="http://xmlns.oracle.com/adf/svc/types/" xmlns:tns="http://
xmlns.oracle.com/apps/crmCommon/content/outboundMessage/types/"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <import namespace="http://xmlns.oracle.com/adf/svc/types/"
    schemaLocation="BC4JService.xsd"/>
  <import namespace="$OBJECT_TARGET_NAMESPACE$"
    schemaLocation="$OBJECT_NAME$.xsd"/>
  <import namespace="http://xmlns.oracle.com/adf/svc/errors/"
    schemaLocation="ServiceException.xsd"/>
  <element name="processOutboundMessage">
    <complexType>
      <sequence>
        <element name="object" type="ns1:$OBJECT_NAME$"/>
      </sequence>
    </complexType>
  </element>
  <element name="processOutboundMessageResponse">
    <complexType>
      <sequence/>
    </complexType>
  </element>
</schema>
```

Security Policy

The object workflow outbound messages can use either an authentication-only client side security policy or a transport-level security policy that can protect the message during transfer. The default authentication-only policy used by object workflow outbound message is oracle/wss10_saml_token_client_policy, which includes security assertion markup language (SAML) tokens in outbound simple object access protocol (SOAP) request messages. This policy should only be used when the target web service is located within a secure network segment. The corresponding service can use any compatible service policy, such as oracle/wss10_saml_token_service_policy or oracle/wss_saml_or_username_token_service. You must protect the message when the service is outside your firewalls. When protected, the outbound message uses oracle/wss_username_token_over_ssl_client_policy with one-way SSL enabled by default. The corresponding service must use the compatible oracle/wss_saml_or_username_token_over_ssl_service_policy service policy.

For more information on basic administration involving deployment, configuration, and management of Web services, see part II of the Oracle Fusion Middleware Security and Administrator's Guide for Web Services.

E-Mail Templates: Explained

You can create, manage, and use e-mail templates for configuring e-mail notifications as part of object workflows. E-mail templates define the layout of the e-mails, which ensures that e-mail notifications triggered by the same type of business event for a specific object have a consistent look and feel. You must have administrative privileges to define e-mail templates.

A template is created for a particular type of object and can be used only with that object when defining e-mail notifications. You can compose e-mail templates

using the Oracle Fusion CRM Application Composer user interface or use any e-mail editor to create a template and then upload the template using the user interface. You can use e-mail templates, for example, for including your company letterhead in outbound e-mail communications.

You can **Search** an e-mail template using either or all of the search parameters such as full or partial **Name** of the template, the **Object** to which the template is associated, and whether the template is **Active**.

Managing E-Mail Templates

Managing e-mail templates include tasks that enable you to **Search**, **View**, **Duplicate**, **Create**, **Edit**, or **Delete** a template.

Options available for managing e-mail templates include:

- Viewing the existing e-mail templates and customizing views.
- Filtering or querying existing templates including partial search using wildcard.
- Editing an existing template or creating a duplicate.

Note

You cannot change the object for which a template is defined.

-
- Creating and configuring a new template.
 - Detaching the listed templates for a full-page view.
 - Viewing which templates are **Active**.

Note

Only an **Active** template can be used for object workflow definition. Saving does not automatically make the template **Active**.

Configuring E-Mail Templates

You can define a template using the formatting features as available in any HTML based e-mail client. Templates are automatically converted to plain text for users who cannot view HTML e-mails. Advanced HTML users can edit the templates using the **Source Code Editing Mode** where the tags are set to visible. These advanced users can copy and edit the source code in any HTML editor, and then paste the edited code back to Oracle Fusion. While defining an e-mail template, the fields you select and embed in a template are specific to the selected object. The fields are populated with values at runtime.

Options available for configuring e-mail templates include:

- Specifying **Name** and **Description** for the template.
- Attaching artifacts relevant to the e-mail template. **Attachments** are included in every e-mail that uses the template having an attachment.

You can also send attachments as links to avoid crowding the Inbox of the recipients. An attachment can be a file on a local computer or a shared file in a repository.

Note

File attachments cannot exceed 10 MB.

-
- Specifying whether an e-mail template is **Active**.

Note

You can use only the **Active** templates when creating object workflows for e-mail notifications.

-
- Inserting the **Fields** or **Functions**, as applicable. You can use **Functions** to insert **Date**, **Current date and time**, or a **Hyperlink to record**, which triggers an e-mail notification. You can also **Browse** and then insert the content of a local file in the body of the e-mail template.
 - Performing basic formatting such as font format, character format, paragraph alignment, bullets and numbering, and so on as in any HTML based e-mail client.

FAQs for Creating Object Workflows

Why are some e-mail templates invalid?

E-mail templates can become invalid if the corresponding template file of the e-mail template cannot be found in the file repository. This can happen if the template file was migrated elsewhere, or deleted accidentally. Contact your application administrator to attempt to recover the template file by reviewing your file repository setup. If the file cannot be recovered, then you can delete the e-mail template and create a new one.

Application Composer: Building Custom Subject Areas

Custom Subject Areas: Explained

A report subject area is a set of objects and fields that represent information about the areas of an organization's business. You select a report subject area from within the Oracle Business Intelligence (BI) Composer when building reports. Such reports display only those records that meet the criteria that is defined as part of the BI Composer report creation flow. To build reports, use either the predefined report subject areas that are delivered for an Oracle Fusion CRM application, or create new ones using a wizard. The report subject areas that you create using the wizard are called custom subject areas.

Review these aspects of custom subject areas, before you begin to create your own custom subject areas:

- Creating custom subject areas
- Editing custom subject areas

Creating Custom Subject Areas

Using the available wizard, you create a custom subject area by selecting a primary object, related objects, and specific fields. When you later build a report within the BI Composer, the custom subject area that you choose as the basis for the report controls the data displayed on the report.

To access the custom subject area wizard:

1. Select **Application Composer** from the Navigator menu, under the Tools category.
2. On the main Overview page, select an application from the **Application** choice list.
3. Select the **Custom Subject Areas** link in the Common Setup pane, or in the local area of the main Overview page.

Considerations when creating custom subject areas:

- Enter a meaningful label and description for the custom subject area. Subject areas usually have names that correspond to the type of information that they contain, such as service requests and orders.

The label identifies the subject area when building or running a report from within the BI Composer.

- All custom subject areas that you create are based on a primary object. The primary object is the top-level reportable object that is the focus of the report that you eventually build using this subject area. If you want your report to include data from these objects, you can also select related objects. For example, quotes could be a primary object, and an opportunity could be a related object.

When selecting related objects for your custom subject area, consider the following points:

- For a one-to-many primary-child relationship, you cannot add more than three child objects for a primary object.
- For a many-to-one primary-child relationship, you can select only one child object.
- For a reference relationship that a child object has with a primary object or any other object, you can select multiple child objects.
- When you select the primary object and related objects, if any, you then add fields to the custom subject area and select the measures for those fields, as applicable.

Tip

You can either select measures for each currency, date, or numeric field individually, or select all such fields at once using the Actions list.

-
- Use date leveling so that you can divide time periods into smaller units, for example, to view information by year or quarter and then by month.
 - Configure the security level that determines which BI duty roles can use the custom subject area for building and viewing reports using the BI Composer.
 - Review the graphical representation of the custom subject area including the objects, their fields, and the measures in the custom subject area.
 - Submit the custom subject area.

You can create and submit a custom subject area, either immediately, or save and close the custom subject area at any point and submit it later.

Note

You must first submit a custom subject area before you can select it from within BI Composer.

Once you save or submit a custom subject area, you cannot modify its primary object.

Editing Custom Subject Areas

Using either the name, description, or primary object, you can search for an existing custom subject area and then modify it.

Note

You cannot modify a predefined report subject area that is delivered with an Oracle Fusion CRM application. Instead, create new custom subject areas to meet your reporting needs. Before you create a new custom subject area, be sure to review all the included subject areas to see if the one you want is already available.

Date Leveling: Explained

Date leveling allows the user to view the data by different levels of time, for example, by year or quarter, and by month.

To configure date leveling, use the Configure Date Leveling step of the Wizard to either allow or disallow leveling. You may have to expand the field list in the **Date** field to select or clear the Date Leveling check box, as applicable.

Measures: Explained

Measures are a set of functions that you can apply on the date, numeric, or currency type fields of the selected primary or child objects while defining a custom subject area. Measures enable you to apply aggregation rules on the report data for a customized view using Oracle Business Intelligence (BI) Composer.

Measures available to a particular type of field may differ depending on the field type. Use actions list to select or clear measures all at once for the applicable fields. Once you define a measure for the desired fields and publish a custom subject area, you can view and select these fields along with the applied measures when defining the report data in the Oracle Business Intelligence Composer.

Here are some measures you can apply to numeric, currency, or date type fields.

- For numeric and currency fields, a measure can be:
 - Sum
 - Average
 - Count
 - Count Distinct
 - Maximum
 - Minimum
 - First
 - Last
 - Median
 - Standard Deviation Sample

- Standard Deviation Population
- For date fields, a measure can be:
 - Maximum
 - Minimum

For more information about selecting the data for a report using the Oracle Business Intelligence Composer, see Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

You can select measures based on your reporting needs. For example, you can use measures to view product sales per store, state, or country. Or, to view the number of support tickets opened or closed per day, week, or month, and so on.

Securing Custom Subject Areas: How It Works

You can secure a custom subject area by granting or revoking access rights from role names, which determines which role names can or cannot access a custom subject area. You can also add more role names from a predefined list and assign or revoke permissions. This topic covers how you can add or delete role names, or grant or revoke access rights from those role names.

Managing Role Names and Access Rights

While defining a custom subject area using the wizard, you can use the actions list in the configure security step to manage role names and access rights as follows:

- Select and add role names for a custom subject area from a predefined list of role names. This predefined list also provides the description for each role name. You can also select and add multiple role names from this predefined list using either the Shift or Ctrl keys. Once you add a new role name, you can select appropriate access for that role name.
- Select and delete role names listed for a custom subject area. You can also select and delete multiple role names using either the Shift or Ctrl keys.

Note

You cannot delete the role name listed as everyone.

-
- Read access is granted by default to each role name you add. If you want to revoke read access from a listed role name, select no access for that role name.

Note

You can create custom subject areas even for the objects in which you do not have access to the data, which allows you to build custom subject areas with compromising data security.

Extension Dimension Reports: Explained

For key objects in CRM Applications, Oracle has provided a pre-seeded extension dimension that includes custom or extended attributes. This functionality enables users to create reports on extensions that are made to standard objects. Extension dimension attributes are not available for reporting until custom attributes have been specified. When expanding the Opportunity extension dimension, only the Opportunity type attribute is displayed.

When you create a custom field, you can create reports for the following data types:

- Boolean
- Number
- Currency
- Date
- String
- Percentage
- Phone
- DateTime

To create extension dimension reports, navigate to CRM Composer. In the **Navigator** menu, click **Application Composer**, and create custom fields for standard objects. Ensure that the custom fields are exposed on the UI, so that users can enter data for these new attributes. Then navigate to Oracle Business Intelligence Composer. In the **Navigator** menu, click **Reports and Analytics**, select a real-time or OTBI subject area that includes the pre-seeded extension dimension, and create a new report.

When you select report columns, you can see that the custom fields now show up under the extension dimension folder.

Publishing a Custom Subject Area: Explained

Publishing makes the data that you configured for a custom subject area available in the Oracle Business Intelligence (BI) Composer for building and viewing reports. Publishing processes begin when you submit a custom subject area in the last step of the custom subject area configuration wizard. This topic covers what happens when you submit a custom subject area for publishing, and what the publishing statuses indicate.

Understanding Publishing

Once you submit a custom subject area for publishing, two processes occur in the background. First process is synchronous and creates Oracle Applications

Development Framework (Oracle ADF) artifacts; hence, you must wait before the first process is over. The second process is asynchronous and creates centralized metadata repository (RPD) fragments and publishes them to the Oracle BI server.

Note

You must refresh the status to know whether the custom subject area is successfully published. You may have to refresh the status multiple times as the Oracle ADF and RPD artifacts creation may require longer time.

Once the status is changed to OK, which means that the custom subject area is published, you can go to Oracle BI Composer and create reports using the fields and functions that you configured for that subject area.

Publishing status for a custom subject area can be as follows:

- **Pending:** When you save and close a custom subject area at any point before submitting. You can return to the pending custom subject area to complete the configuration process. This status may also indicate a failure in the background processes while creating Oracle ADF and RPD artifacts for a custom subject area.
- **In Process:** When you submit a custom subject area, but the submitted data is in the process of being published to the Oracle BI.

Note

If the in process status does not change to OK even after multiple refresh attempts, there could be an error in publishing.

-
- **OK:** When a custom subject area is successfully published and is available to the Oracle BI Composer for report configuration and analyses.

Note

You can edit a custom subject area in any status and resubmit.

For more information on using the Oracle BI Composer for creating and viewing a report, see Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition).

FAQs for Building Custom Subject Areas

What's a primary object?

The primary object is the focus of a custom subject area, which you can later select in Oracle Business Intelligence (BI) Composer to build a report. The list of values for the primary object includes all top-level, reportable objects that belong to the application you selected to customize using the Oracle Fusion CRM Application Composer. The list of values does not include common

component objects, such as notes, interactions, or tasks. Reportable objects are either top-level, custom objects or standard objects that have been configured to be reportable by the owning Oracle Fusion CRM application.

Note

Once you select the primary object and save the custom subject area, you cannot change the primary object.

Can I change a custom subject area's primary object?

No. Once you save a custom subject area, you cannot change its primary object; however, you can create a new custom subject area with a different primary object.

What's a child object?

A child object is related to a primary or parent object and can be a primary or parent object of another child object. For example, an opportunity could be a parent object and its child objects could be an opportunity contact, opportunity reference, opportunity revenue, and so on. Such a relationship defines how objects in a database are related. While a primary object is the focus of the custom subject area, its related child object can be used to add further details to the report. Depending on whether the primary-child relationship is one-to-many or many-to-one, the permissible number of child objects that can be added to a primary object differ while defining a custom subject area.

What's a field?

A field is an object in the database that you can select for defining a custom subject area. The fields available for selection during the custom subject area task flow are determined based on the relationship between the primary and child objects.

What happens if I change a custom subject area after it is published?

You can edit a published custom subject area and then republish it once your changes are done. Modifying a custom subject area does not affect the reports that you created using that custom subject area before making the changes. You can use the modified custom subject area should you need to enhance existing reports.

Note

You cannot edit a primary object when you modify a custom subject area. Should you need to do so, create a new custom subject area using a different (new) primary object.

Oracle Composer: Customizing Oracle Fusion CRM Applications

Customizing Oracle Fusion CRM Pages Using Oracle Composer: Explained

You can customize most pages in Oracle Fusion CRM using Oracle Composer. To access Oracle Composer, navigate to one of the pages available for customization and select to customize the page from the **Administration** menu in the global area.

Available Customization Options

When you select to customize a page from the **Administration** menu in the global area, you launch Oracle Composer.

The customizations that you make are applied based on your layer selection:

- Site
Your customizations are visible to all users.
- Job Role
Your customizations are visible to users who have the selected job role.
- External or Internal
Depending on your selection, your customizations are visible to either external or internal users. External users could be your partners or anonymous users. Internal users could be your employees.

Note

Users can customize their own dashboard pages by selecting Edit Current Page from the Personalization menu. Such customizations are visible only to the user who made the change.

Available Customization Tasks

Oracle Composer uses two different modes of Design View. The first mode, Design View: Standard mode, is selected by default in all CRM pages when opening a page with Oracle Composer with the Design tab selected. The second

mode, Design View: Direct Selection mode, is activated when you click the Select tab for the UI page you want to customize.

With the Design View: Direct Selection mode, you can select and edit UI components such as form fields and table columns. In Direct Selection mode, selectable UI components become apparent when you move your cursor over the UI component. Selectable UI components are highlighted and can be edited. In CRM, Direct Selection mode is available when you customize pages, but not when you personalize a dashboard page.

Examples of UI components exposed in Design View: Direct Selection mode are:

Fields (for example an input text field)

- Show component (boolean attribute)
- Read only (boolean attribute)
- Required (boolean attribute)
- Label

UI container (for example, form container)

- Show or hide child components (for example, fields in the form)
- Reorder child components (for example, reorder fields in a form)

UI components can be protected from update to preserve the product business logic (for example, a field is read only or not based on a security privilege already defined in the application). If protected from update, the UI components are grayed out and cannot be changed.

This table describes how you can use each mode of Oracle Composer to customize dashboard pages and other select pages*.

Customization Tasks	Design View - Standard mode	Design View - Direct Selection mode
Add content (Business Intelligence reports, CRM portlets such as Calendar)	Yes	No
Delete region	Yes	No
Move region	Yes	No
Change page layout (for example, change a two column layout to three column layout)	Yes	No

*Examples of such pages are: Partner Public Profile page, Partner Landing page, Partner Registration page, Opportunity Overview, Customer Snapshot node, and Overview Analysis tab.

This table describes how you can use each mode of Oracle Composer to customize transactional pages (all other non-dashboard pages).

Customization Tasks	Design View - Standard mode	Design View - Direct Selection mode
Default region state (open or close)	Yes	No

Manage saved searches (create and edit)	Yes	No
Hide or show field	No	Yes
Change field label	No	Yes
Make field required or not	No	Yes
Make field read-only or updateable	No	Yes
Reorder fields in a form	No	Yes
Reorder table columns	Yes	Yes
Hide or show table columns	Yes (end users can optionally display columns at runtime)	Yes (end users cannot display columns at runtime)
Set table column width with the mouse	Yes	No
Set table column width and minimum width in percent or pixels	No	Yes
Make column sortable or not	No	Yes

Customization Privileges Required

The following are examples of job roles that must be assigned to you before you can make customizations using Oracle Composer:

Application	Duty Role	Job Role
CDH	Master Data Management Application Administrator	Master Data Management Application Administrator
Sales Catalog	Sales Catalog Administrator Duty	Sales Catalog Administrator
Sales	Sales Administrator Duty	Sales Administrator
Marketing	Marketing Operations Manager Duty	Marketing Operations Manager
PRM	Channel Administrator Duty	Channel Administrator
PRM	Channel Partner Portal Administrator Duty	Channel Partner Portal Administrator

Contact your security administrator for details.

Customizing Partner Management Pages Using Oracle Composer: Explained

In Oracle Fusion Partner Management, you can customize a variety of pages using Oracle Composer. To access Oracle Composer, navigate to one of the Partner Management pages that is enabled for customization, and select to customize that page from the **Administration** menu or **Personalization** menu in the global area. To customize the Partner Portal user interface (UI) shell, however, the Channel Partner Portal Administrator must use the customization links available in the Tasks pane on the Channel dashboard.

Available Customization Options

When you select to customize a page from the **Administration** menu in the global area, or from the Channel dashboard for certain pages, you launch Oracle Composer.

The customizations that you make are applied based on your layer selection:

- Site
Your customizations are visible to all users.
- Job Role
Your customizations are visible to users who have the selected job role.
- External or Internal
Depending on your selection, your customizations are visible to either external or internal users. External users could be your partners or anonymous users. Internal users could be your employees.

Note

Users can customize their own dashboard pages by selecting **Edit Current Page** from the **Personalization** menu. Such customizations are visible only to the user who made the change.

Oracle Composer uses two different modes of Design View. The first mode, Design View: Standard mode, is selected by default in all CRM pages when opening a page with Oracle Composer with the Design button selected. The second mode, Design View: Direct Selection mode, is activated when you click the Select tab for the UI page you want to customize.

With the Design View: Direct Selection mode, you can select and edit UI components such as form fields and table columns. In Direct Selection mode, selectable UI components become apparent when you move your cursor over the UI component. Selectable UI components are highlighted and can be edited. In CRM, Direct Selection mode is available when you customize pages, but not when you personalize a dashboard page.

Note

Source View is also available to administrators with the Channel Partner Portal Administrator job role. Administrators use this view to make changes to the Partner Portal UI shell.

This table lists the types of customizations that are available for Partner Management dashboard pages and other select pages. You perform all customizations in Design View.

Oracle Fusion Partner Management Page	Customization Task
<ul style="list-style-type: none">• Channel dashboard• Partner dashboard• Other select pages, such as the Partner Public Profile page, Partner Landing page, and Partner Registration page	Add new content from the Resource Catalog.

<ul style="list-style-type: none"> • Channel dashboard • Partner dashboard • Other select pages, such as the Partner Public Profile page, Partner Landing page, and Partner Registration page 	Delete a region from the dashboard. <hr/> Note You can later add the region back to the dashboard using the Resource Catalog.
<ul style="list-style-type: none"> • Channel dashboard • Partner dashboard • Other select pages, such as the Partner Public Profile page, Partner Landing page, and Partner Registration page 	Change the page layout (for example, change from a two column layout to a three column layout).
<ul style="list-style-type: none"> • Channel dashboard • Partner dashboard • Other select pages, such as the Partner Public Profile page, Partner Landing page, and Partner Registration page 	Move regions on a page: <ul style="list-style-type: none"> • Move regions by drag and drop • Add content within an existing box (box displayed in dotted lines) • Add a box (content placeholder) up, down, left, or right by clicking the Add Box icons.

Customizing the Partner Portal UI Shell

A user assigned the Channel Partner Portal Administrator job role can execute this customization task. After logging in, the administrator user can customize the UI shell by clicking the Update partner portal UI shell link in the regional task list. This customization task uses a specific Oracle Composer mode called Source View. For more information on Source View, please refer to Oracle Fusion Middleware User's Guide for Oracle WebCenter Spaces.

Since you are customizing the partner portal UI shell for partner users and anonymous users, you make changes at the Internal or External level (using only External).

You can change:

- UI shell header (for example, add new content, hide or show global hyperlinks, and hide, show, or add menu items)
- UI shell footer (for example, add links)
- Branding logo
- Branding text
- Menu display: use either the Fusion navigator or a tabbed style menu

For example, to switch from the Fusion navigator to a tabbed style menu:

- In Oracle Composer, select the **Navigators** component on the UI, which highlights the corresponding element in Source View.
- Hide the **commandLink: Navigators** element by deselecting the **Show Component** property.
- Show the **panelGroupLayout: horizontal** element by selecting the **Show Component** property.

Customization Privileges Required

To customize the Partner Portal UI shell, your assigned job role must also include the Partner Portal Customize Links Duty duty role. This duty role is assigned to the Channel Partner Portal Administrator job role by default. To customize the Channel or Partner dashboard, you must have the Channel Administrator job role.

FAQs for Customizing Oracle Fusion CRM Applications

What's the difference between Oracle Composer and Oracle Fusion CRM Application Composer?

Oracle Composer is an Oracle Fusion tool you can use to modify Oracle Fusion user interface (UI) pages and components for all products designated for use with Oracle Composer. Oracle Composer uses two different modes of Design View. The first mode, Design View: Standard mode, is selected by default in all CRM pages when opening a page with Oracle Composer with the Design button selected. The second mode, Design View: Direct Selection mode, is activated when you click the Select tab for the UI page you want to customize. In CRM, Direct Selection mode is available when you customize pages, but not when you personalize a dashboard page. With the Design View: Direct Selection mode, you can select and edit UI elements such as form fields and table columns. In Direct Selection mode, selectable UI components become apparent when you move your cursor over the UI component. Selectable UI components are highlighted and can be edited.

This table describes how you can use each mode of Oracle Composer to customize dashboard pages and other select pages (such as the Partner Public Profile page, Partner Landing page, Partner Registration, Customer Snapshot, and Customer Overview - Analysis tab), and customize transactional pages (all other non-dashboard pages):

Use Cases	Design View - Standard mode	Design View - Direct Selection mode	Page Type
Add content (Business Intelligence reports, CRM portlets such as Calendar)	Yes	No	Dashboard and other select pages
Delete region	Yes	No	Dashboard and other select pages
Move region	Yes	No	Dashboard and other select pages
Change page layout (for example, change a two column layout to three column layout)	Yes	No	Dashboard and other select pages
Default region state (open or close)	Yes	No	Transactional pages (all non-dashboard pages)
Manage save queries (create and edit)	Yes	No	Transactional pages (all non-dashboard pages)

Hide or show field	No	Yes	Transactional pages (all non-dashboard pages)
Change field label	No	Yes	Transactional pages (all non-dashboard pages)
Make field required or not	No	Yes	Transactional pages (all non-dashboard pages)
Make field read-only or updateable	No	Yes	Transactional pages (all non-dashboard pages)
Reorder fields in a Form	No	Yes	Transactional pages (all non-dashboard pages)
Reorder table columns	Yes	Yes	Transactional pages (all non-dashboard pages)
Hide or show table columns	Yes	Yes	Transactional pages (all non-dashboard pages)
Set table column width with the mouse	Yes	No	Transactional pages (all non-dashboard pages)
Set table column width and min width in percent or pixels	No	Yes	Transactional pages (all non-dashboard pages)
Make column sortable or not	No	Yes	Transactional pages (all non-dashboard pages)

The Oracle Fusion CRM Application Composer also lets you make UI changes at runtime. However, the types of UI changes that you can make using the Application Composer are quite different. Specifically, your primary focus when using the Application Composer is to make actual object model changes. For example, you can create a new business object and related fields, and then create new application pages where that object and its fields are exposed to users. The ability to make these types of object model extensions is available only in Oracle Fusion CRM applications. Also, using the Application Composer, you cannot access the Resource Catalog to add new content to a page. With Application Composer, administrators can make customizations at the site level only.

This table describes some of the primary differences between Oracle Composer and the Application Composer:

Customization Task	Available in Oracle Composer (site, job role, external or internal level)?	Available in Oracle Fusion CRM Application Composer (site level only)?
Make object model extensions and expose your customizations by creating or modifying work area pages	No	Yes
Reorder subtabs	No	Yes
Customize dashboard pages	Yes	No
Add content from the Resource Catalog	Yes	No
Simple field customizations (show, hide, make read only, make required)	Yes (WYSIWYG - what you see is what you get)	Yes (non-WYSIWYG)

Select the MDS layer where you want to author customizations, such as at the site layer or job role layer	Yes	No
View results of customizations immediately	Yes, in the Oracle Composer design interface	Yes, in the CRM application that you are customizing

Glossary

BPEL

Business Process Execution Language; a standard language for defining how to send XML messages to remote services, manipulate XML data structures, receive XML messages asynchronously from remote services, manage events and exceptions, define parallel sequences of execution, and undo parts of processes when exceptions occur.

business object

A resource in an enterprise database, such as an invoice or purchase order.

custom subject area

A custom subject area is a set of records and fields that represent information about the business areas of an organization. A custom subject area includes a primary object, on which a report is focused, along with its child objects and their fields that you use for creating and viewing reports.

Measure

A measure is a function, such as sum, average, or median, that is available for fields of type, date, currency, or numeric. A measure is used for aggregating report details.

object workflow

Object workflows represent a series of automated tasks configured for a business object, which are invoked based on a set of trigger conditions.

primary object

A primary object is a top-level, reportable object that may contain one or more child objects.

WSDL

Abbreviation for Web Services Description Language. It is an XML format that provides a model for describing Web services.