Oracle

Engagement Cloud
Implementing Service in Engagement Cloud

Release 13 (update 18B)
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Preface</strong></td>
<td>i</td>
</tr>
<tr>
<td><strong>1 About This Guide</strong></td>
<td>1</td>
</tr>
<tr>
<td>Audience and Scope</td>
<td>1</td>
</tr>
<tr>
<td>Related Guides</td>
<td>1</td>
</tr>
<tr>
<td><strong>2 Understanding Setup Tasks</strong></td>
<td>3</td>
</tr>
<tr>
<td>Understanding Implementation Structures</td>
<td>3</td>
</tr>
<tr>
<td>Managing an Implementation</td>
<td>4</td>
</tr>
<tr>
<td><strong>3 Using Profile Options, Lookups, and Scheduled Processes</strong></td>
<td>9</td>
</tr>
<tr>
<td>Profile Options, Lookups, and Scheduled Processes: Overview</td>
<td>9</td>
</tr>
<tr>
<td>Profile Options: Explained</td>
<td>9</td>
</tr>
<tr>
<td>Lookup Types: Explained</td>
<td>10</td>
</tr>
<tr>
<td>Modifying Service Request Lookups: Explained</td>
<td>11</td>
</tr>
<tr>
<td>FAQs for Lookups and Lookup Types</td>
<td>12</td>
</tr>
<tr>
<td>Scheduled Processes: Explained</td>
<td>13</td>
</tr>
<tr>
<td>Profile Options and Scheduled Processes for SR Management</td>
<td>14</td>
</tr>
<tr>
<td><strong>4 Setting Up Users and Security</strong></td>
<td>19</td>
</tr>
<tr>
<td>Defining Setup Users: Overview</td>
<td>19</td>
</tr>
<tr>
<td>Setting Up Users and Security: Overview</td>
<td>19</td>
</tr>
<tr>
<td>About Resources and Resource Management: Explained</td>
<td>22</td>
</tr>
<tr>
<td>Resource Team Information</td>
<td>23</td>
</tr>
<tr>
<td>FAQs for Define Resource Team Information</td>
<td>24</td>
</tr>
<tr>
<td>About Security Roles: Explained</td>
<td>25</td>
</tr>
<tr>
<td>Setting Up Service Request Visibility Based on Queue</td>
<td>26</td>
</tr>
<tr>
<td>Setting Up Service Request Visibility Based on BU</td>
<td>28</td>
</tr>
</tbody>
</table>
5 Managing Catalogs

Defining a Catalog for the Service Offering: Explained
Sales Catalogs: Overview
Creating the Sales Catalog: Getting Started
Adding Products to the Catalog: Procedure
Enabling the Sales Catalog
Validating the Service Catalog

6 Setting Up Work Assignment and Routing

Work Assignment: Implementation Concepts
Assignment Mappings
Assignment Rules
Categories
Managing Work Assignment
Enabling Omnichannel
FAQs About Service Requests

7 Setting Up Communication Channels

Inbound Message Processing: Explained
Setting Up Email Channels: Procedure
Inbound Message Filters: Explained
Creating and Updating Inbound Message Filters: Procedure
Defining Email Templates: Procedure
Configuring Email Profile Options: Procedure
Configuring an Email Channel: Procedure
Configuring a Job to Process Inbound E-mails: Procedure
Enabling Service Request Outbound E-Mail: Explained
Modifying Acknowledgment Messages for Inbound Email
Setting Up Social Channels: Explained
Integrating Engagement Cloud with Social Cloud: Procedure
Engagement Cloud to Social Cloud Integration Mapping: Explained
Configuring the Cross-Channel Interactions: Procedure
Using Groovy Scripts for Social Channel: Explained
Setting Chat Profile Options: Procedure
FAQs About Communication Channels
8 Configuring Computer Telephony Integration (CTI)

Computer Telephony Integration (CTI): Explained
Configuring the Media Toolbar: Procedure
Configuring Live Window and Companion Tab for CTI
Configuring Business Objects: Procedure
Configuring Screen Pop Pages: Explained
Configuring Screen Pop Pages: Procedure
Creating Lookup Filters: Procedure
Configuring Call Flow Parameters: Procedure
Modifying the Default Contact Search: Procedure

9 Understanding CTI Media Toolbar APIs

Media Toolbar API: Overview
Media Toolbar API: Methods
Media Toolbar API: Introduction
Configuration APIs: Explained
Interaction API: Explained
Window Management API: Explained
Event Listener API: Explained
API Features: Explained
System Tokens: Explained
System Pages: Explained
System Business Objects: Explained
Channels Code: Explained
Application Classification Code: Explained
Error Messages: Explained
Channel Type Data: Explained

10 Setting Up Service Entitlements

Service Entitlements Overview
Setting Up Milestones and Coverages
Configuring Milestones and Coverages
## Setting Up Action Plans 153

- Action Plan: Overview 153
- Managing Action Plan Actions 154
- Managing Action Plan Templates 156
- Managing Mapping of Action Plan Status Values 157

## Enabling Productivity Tools 159

- Enabling Productivity Tools for Service Requests: Explained 159
- Managing SmartText Entries: Explained 160
- Managing Keyboard Shortcuts: Explained 161

## Setting Up Social Networking 165

- Social Networking: Overview 165
- Enabling Oracle Social Network Objects for Service Requests: Explained 165

## Setting Up Service Business Units 167

- Setting Up Business Units for Service: Overview 167
- Managing Common CRM BU Profile Options for Service BU 168
- Managing Internal Resource Organizations for Service BU 168
- Managing Resource Organization Hierarchies for Service BU 169
- Creating a Business Unit for Service BU 169
- Associating Resource Organizations to Business Units for Service BU 170
- Creating Employees for a Service BU 170
- Working with the Scope in Service BU Setup 171
- Completing the Setup Tasks for Service BU 172
- Securing the Business Unit Field on a Service Request 173
- Exporting and Importing the Functional Setup Data for Business Units 174
- Setting Up Service Request Visibility Based on BU: Overview 176

## Understanding Service Analytics 177

- Implementing Service Analytics: Explained 177
- Enabling the Service Analytics Icon on the Springboard 177
- Roles for Service Analytics 178
- Subject Areas for Service Analytics 179
- Saved Queries for Service Analytics 181
# Understanding Modifications and Extensibility

16

- Extending Oracle Cloud Applications: Explained
- Extending the Service Request Layout in Mobile: Explained
- Setting Up and Defining Groovy Notification Triggers: Explained
- Setting up Service Requests for Partners: Explained

## Importing and Exporting

17

- File-Based Data Import and Export: Overview
- Bulk Export: Overview

## Integrating Oracle Field Service Cloud

18

- Oracle Field Service Cloud: Overview
- Integration Component Architecture Between Oracle Engagement Cloud and Oracle Field Service Cloud: Explained

## Integrating Oracle Internet of Things Service Monitoring for Connected Assets

19

- Oracle Internet of Things Service Monitoring for Connected Assets: Overview
- Integrating Engagement Cloud with Internet of Things Service Monitoring for Connected Assets: Explained

## Implementing Digital Customer Service

20

- Overview of Digital Customer Service
- About Digital Customer Service Terminology
Preface

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

Using Oracle Applications

Using Applications Help

Use help icons ? to access help in the application. If you don’t see any help icons on your page, click your user image or name in the global header and select Show Help Icons. Not all pages have help icons. You can also access Oracle Applications Help.

Watch: This video tutorial shows you how to find help and use help features.

You can also read Using Applications Help.

Additional Resources

- **Community**: Use Oracle Cloud Customer Connect to get information from experts at Oracle, the partner community, and other users.

- **Guides and Videos**: Go to the Oracle Help Center to find guides and videos.

- **Training**: Take courses on Oracle Cloud from Oracle University.

Conventions

The following table explains the text conventions used in this guide.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates user interface elements, navigation paths, or values you enter or select.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates file, folder, and directory names, code examples, commands, and URLs.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than symbol separates elements in a navigation path.</td>
</tr>
</tbody>
</table>

Documentation Accessibility

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

Videos included in this guide are provided as a media alternative for text-based help topics also available in this guide.
Contacting Oracle

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

Comments and Suggestions
Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com.
1 About This Guide

Audience and Scope

This guide is intended for you if you’re responsible for implementing Service Request Management in Oracle Engagement Cloud.

This guide does not cover the implementation activities for Oracle Sales Cloud.

To set up and work with the additional features of Oracle Sales Cloud, see Oracle Sales Cloud documentation on Oracle Help Center at https://docs.oracle.com.

Related Guides

To understand more about the implementation tasks covered in this guide, you can refer to the following table for a list of related guides.

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Engagement Cloud Using Service in Engagement Cloud</td>
<td>Contains information to help service managers, service personnel, and other service end users to perform day-to-day business tasks using Oracle Cloud.</td>
</tr>
<tr>
<td>Using Knowledge in Engagement Cloud</td>
<td>Describes how administrators, agents, authors, and other knowledge base contributors can implement and use Knowledge in Engagement Cloud.</td>
</tr>
<tr>
<td>Oracle Engagement Cloud Integrating Engagement Cloud with Field Service Cloud</td>
<td>Outlines the implementation and configuration steps required to integrate, create, and update processes on service work orders in Oracle Engagement Cloud with activities in Oracle Field Service Cloud.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Getting Started with Your Sales Implementation</td>
<td>Describes your initial Oracle Sales Cloud service implementation procedures, based on a simple sales-force-automation use case.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Implementing Customer Data Management</td>
<td>Contains information to help implementors define the setup for managing customer information and the configuration for customer hub deployment.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Implementing Enterprise Contracts</td>
<td>Contains conceptual information and procedures needed to implement the contract management features of Oracle Sales Cloud.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Implementing Marketing</td>
<td>Contains conceptual information and procedures needed to implement the marketing components and features of Oracle Sales Cloud.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Implementing Sales</td>
<td>Contains conceptual information and procedures needed to implement components and features of Oracle Sales Cloud.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Understanding File-Based Data Import and Export</td>
<td>Contains information to help those charged with exporting and importing object data.</td>
</tr>
<tr>
<td>Title</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Oracle Sales Cloud Securing Sales</td>
<td>Contains information to help setup users and sales administrators configure access to Oracle Sales Cloud functionality and data.</td>
</tr>
<tr>
<td>Oracle Sales Cloud Security Reference</td>
<td>Lists the predefined security data that is included in the Sales offering.</td>
</tr>
</tbody>
</table>

**Related Topics**

- [Oracle Help Center](#)
2 Understanding Setup Tasks

Understanding Implementation Structures

Functional Setup Manager: Overview

Oracle Functional Setup Manager provides an integrated, end-to-end process for functional administrators to manage the implementation and maintenance of Oracle Applications Cloud. Functional Setup Manager offers the following:

- Standardized application configuration and setup experience
- Feature opt-in for a best fit configuration
- Flexible processes for managing setup:
  - Setup by functional areas for an adopt-as-you-go approach
  - Implementation projects to manage setup
  - Upload file to enter setup data in bulk
- Guided task list for end-to-end setup requirements
- Export and import services for setup data migration between environments
- Comprehensive reporting on setup data

Planning Your Implementation: Explained

To plan for the implementation of your subscribed Oracle Applications Cloud, identify the offerings you need to implement. Functional Setup Manager provides documentation to help you to understand the functionality and the setup requirements of the offerings. Review those documents and then prepare the data you need to implement the relevant offerings.

Functional Setup Manager Components: How They Work Together

Offerings, functional areas, features, and setup tasks work together in your implementation.

Offerings

An offering represents a collection of business processes that are supported by Oracle Applications Cloud. Each subscription of Oracle Cloud provides license to use one or more offerings and they’re the starting point of all implementations. An offering consists of multiple functional areas and features.

Functional Areas

A functional area represents one or more business sub-processes and activities within its parent offering. It may represent a core operation of the offering or may represent an optional activity which may or may not be applicable to your business. When you start to implement an offering by enabling it, core functional areas are enabled automatically. You have a choice
to opt into and enable an optional functional area or to opt out of it. A functional area may be divided into smaller functional areas creating a hierarchy to help you to decide what to opt into one step at a time. Some of the functional areas may be applicable to more than one offering. Once you set up a shared functional area, you do not have to set it up again when implementing another parent offering. However, Oracle recommends that during successive implementation of the other parents you verify if there are any offering-specific tasks that may still require your attention.

Features

Features are optional business practices or methods applicable to the functional areas. Like functional areas, you can decide to opt into or opt out of features depending on the requirements of your business processes. Features can be one of three different types:

- Yes or No: These features allow you either to opt into or to opt out of them and are represented by a single check box. You select them to opt into or deselect them to opt out.
- Single Choice: These features offer multiple choices but allow you to select only one option. Select the option applicable to your business processes.
- Multi-Choice: These features offer multiple choices but allow you to select more than one of the choices. Each choice is presented with a check box. Select all that apply to your business processes by checking the appropriate choices.

Setup Tasks

Setup tasks represent the work necessary to set up an offering and the business processes and activities that the offering represents to make them ready for transaction processing. Perform these tasks to enter setup data when you implement an offering.

Tasks representing setup requirements of the offerings and the functional areas are grouped into task lists and are organized in a hierarchy. For example, all setup tasks of an offering are grouped into a task list which includes subtask lists that represent setup of functional areas within the offering. This helps you gain visibility into setup data that are related to each other, helping you to manage setup.

Managing an Implementation

Enabling Offerings: Explained

Offerings and their functional areas are presented in an expandable and collapsible hierarchy to facilitate progressive decision making regarding whether or not you want to implement them. An offering or its functional areas can either be opted into or not opted into for implementation. Implementation managers decide which offerings to enable for implementation. Although all of the functional areas that represent core functionality of an offering are automatically enabled for implementation when a parent offering is enabled for implementation, you can select which of the optional functional areas are enabled. You can identify which functionality is already opted into by looking at the check box in the Enable column.

Related Topics

- Configuring Offerings
Configuring Offerings: Procedure

Enable offerings to modify functionality so that it matches the services you plan to implement. You need the Configure Oracle Fusion Applications Offering privilege (ASM_CONFIGURE_OFFERING_PRIV) to enable offerings.

Enable Offerings

To enable offerings, follow these steps:

1. Click Navigator > My Enterprise > Offerings work area.
2. In the Offerings page, select the offering you want to implement.
3. Click the Opt In Features button.
4. In the Opt In page, select the Enable check box for the offering.
5. Review functional area hierarchy. Select the Enable check box to opt into functional areas as applicable to your business operations.
6. Click the Features icon in the Features column for the functional area you enabled to opt into and enable applicable features.
   - Depending on the feature type, a check box for Yes or No features or a Features icon for single and multiple choice features is displayed in the Enable column.
   - To enable a feature, select the check box for Yes or No types or click Features and select the appropriate choices for single and multiple choice features.
7. Click Done when you're finished to return to the Opt In page.
8. Click Done to return to the Offerings page.

Repeat the same steps for each offering you want to implement or if you must change the opt-in configuration of any functional areas or features of an enabled offering.

Related Topics

- Configuring Offerings

Adopting New Functionality: Explained

As your business needs change or expand, you may need to adopt new functionality not included in your initial implementation. You can adopt any new functionality for your subscribed offerings to satisfy your business needs. All functionality available for an offering is listed in the Opt In page whether or not you have opted into it. In order to opt into new functionality, make sure that its parent in the hierarchy is already selected. Additionally, Functional Setup Manager provides you easy access to learn more about any feature by clicking the feature's Help icon.

Opting into New Features After Upgrade: Explained

New functional areas and features for an offering you implemented are often introduced in the later revisions of Oracle Applications Cloud. You can use the New Features work area to explore and learn about what has been introduced in the last upgrade of your applications and decide whether to opt into them.

You can review the new functional areas and features of all your enabled offerings or focus on only one of them. For each functional area or feature, you can view its opt-in status, check whether it requires setup, and access additional help topics to learn more details.
Managing Setup Using Offering Functional Areas: Explained

After you enable an offering and configure the opt-in selection of its functional areas and features, you can set up the offering by using its functional areas as a guide. This adopt-as-you-go approach to functional setup gives you the flexibility to set up different functional areas of the offering at different times.

For example, you can begin with setup of the functional areas you require immediately to start transactions. You can then set up other functional areas as you adopt additional offering functionality over time. This setup process is ideal for an enterprise looking for a simpler implementation approach that follows setup best practices.

Functional Areas

When using this method, you start by selecting one of the offerings you enabled. Based on your opt-in configuration, all its enabled functional areas, which include core and optional functional areas, are automatically displayed in a list to guide you through the setup tasks. The display order reflects the sequence in which the functional areas should be set up because setup data of the functional areas listed higher up in the list are usually prerequisite for those shown lower in the list. Any functional area for which setup is mandatory is marked with an asterisk.

Functional areas that are applicable to more than one of your enabled offerings are marked as shared to allow you to evaluate whether they were previously set up during the implementation of another offering. Even if a shared functional area was set up previously, you may still need to evaluate if it requires additional setup data for the offering you are presently implementing.

For some functional areas, Quick Setup may be available to implement its basic functionality quickly. A Quick Setup icon next to a functional area indicates if Quick Setup is available. You can use this task instead of the setup task list to set up those functional areas.

Setup Tasks

For each functional area, a sequenced list of tasks representing the setup best practices according to your opt-in configuration of the features is shown to guide you through optimal implementation requirements. Use the tasks to enter the setup data they represent. Like functional areas, the display order of the tasks always reflects the sequence in which they should be performed to address setup data dependencies.

Required Tasks

Only the required setup tasks are shown by default to minimize your setup effort and to make the offering ready for transactions sooner. However, you can also review the rest of the tasks in the list, which are typically optional or have predefined default values based on common use cases, and decide whether your implementation must change their default setup data.

Tasks with Scope

If any setup data is segmented by a specific attribute or scope, you may need to perform the task iteratively. If so, you must select a qualifying scope value prior to performing the task. You can pick a scope value that was previously selected, select a new scope value, or create a new scope value and then select it. The selected value is a qualifying attribute of the setup data and therefore, different setup data can be entered for the different scope values.
Note: You cannot perform a task if you do not have the proper security privileges.

Related Topics

- Setting up Offerings
- Setting Up Offerings with Scope
- Managing Setup Using Offering Functional Areas: Procedure

Migrating Data Between Environments: Points to Consider

Almost all Oracle Fusion application implementations require moving data from one instance into another at various points in the life cycle of the applications. For example, one of the typical cases in any enterprise application implementation is to first implement in a development or test application instance and then deploy to a production application instance after thorough testing. You use various methods or tools to accomplish the migration of data.

For more information, see the Importing and Exporting Setup Data chapter of the Oracle Applications Cloud Using Functional Setup Manager guide.

Related Topics

- Oracle Applications Cloud Using Functional Setup Manager
- Exporting Offering Setup
- Importing Offering Setup
3 Using Profile Options, Lookups, and Scheduled Processes

Profile Options, Lookups, and Scheduled Processes: Overview

In Oracle Sales Cloud, profile options, lookup types, and scheduled processes let you configure application behavior and refresh data.

Briefly, the following are the purposes of profile options, lookup types, and scheduled processes:

- **Profile options**: Let you configure the application behavior.
- **Lookup types**: Provide the lists of values in applications. Many lookup types can be modified to fit your business needs.
- **Scheduled processes**: Refresh data in the applications.

You can find additional information on profile options, lookup types, and scheduled processes in this chapter and in the related topics.

Related Topics

- How can I access predefined profile options?
- How can I access predefined lookups?
- Viewing Details About Predefined Scheduled Processes: Procedure

Profile Options: Explained

Profile options let you configure and control application data centrally. Administrators and setup users manage profile options in the Setup and Maintenance work area.

Profile options store various kinds of information. The following table lists some examples.

<table>
<thead>
<tr>
<th>Type of Information</th>
<th>Profile Option Setting Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>User preferences</td>
<td>Provides access to social networking features</td>
</tr>
<tr>
<td>Installation information</td>
<td>Identifies the location of a portal</td>
</tr>
<tr>
<td>Configuration choices</td>
<td>Changes UI skins and behaviors</td>
</tr>
<tr>
<td>Processing options</td>
<td>Determines how much information to log</td>
</tr>
</tbody>
</table>
Profile Option Hierarchy Levels

Profile options can be set at different levels, such as site level or user level. The application gives precedence to certain levels over others, when multiple levels are set. The allowed levels come preconfigured with the application.

In the predefined profile option levels, the hierarchy levels and their precedence are:

1. User: This level affects only the current user. It has the highest precedence, over Site and Product.
2. Product: This level affects a product or product family. The application gives it priority over Site level. However, if the user level is set, the user level takes precedence.
3. Site: This level affects all applications for a given implementation. The application gives it the lowest precedence when other levels are set. If no other levels are set, however, it is the highest level.

As a best practice, set site-level profile option values before specifying values at any other level (where available). The profile option values specified at the site-level work as the default until profile option values are specified at the other levels.

The following table shows an example of the predefined profile option hierarchy levels and their priorities.

<table>
<thead>
<tr>
<th>Level</th>
<th>Priority</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site</td>
<td>Lowest</td>
<td>Currency for a site is set to Euros.</td>
</tr>
<tr>
<td>Product</td>
<td>Supersedes Site</td>
<td>Currency for the product or set of products is set to UK pound sterling.</td>
</tr>
<tr>
<td>User</td>
<td>Highest, supersedes Product</td>
<td>Currency for a user is set to US dollars.</td>
</tr>
</tbody>
</table>

You can find additional information on profile options in the related topics.

**Related Topics**

- How can I access predefined profile options?
- Profile Options and Related General Preferences: How They Work Together
- Profile Options: Overview

Lookup Types: Explained

Lookup types provide the lists of values in application fields that are drop-down lists. For example, while creating a service request, service personnel can select the severity of the SR, which is a drop-down list. The values in that list are derived from the lookup type, ORA_SVC_SR_SEVERITY_CD. This lookup type has several potential values known as lookups, each with their own unique lookup code and a meaning that displays in the UI.

Modifying Lookup Types

You can modify many lookup types during or after implementation.

The configuration level of a lookup type determines whether the lookups in that lookup type can be edited. The lookup configuration levels are: User, Extensible, and System.
The following table shows which lookup management tasks are allowed at each modification level.

<table>
<thead>
<tr>
<th>Allowed Task</th>
<th>User</th>
<th>Extensible</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deleting a lookup type</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Inserting new codes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Changing the wording that displays on the page (Meaning field)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Updating start date, end date, and enabled fields</td>
<td>Yes</td>
<td>Yes, only if the code is not predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Deleting codes</td>
<td>Yes</td>
<td>Yes, only if the code is not predefined data</td>
<td>No</td>
</tr>
<tr>
<td>Updating tags</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Updating module</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

If a product depends on a lookup type, the configuration level must be set to system or extensible to prevent deletion. Once the configuration level is set for a lookup type, it can’t be modified. The configuration level for lookup types created using the Define Lookups page is by default set at the User level.

**Service Lookup Types**

You can find lookup types by searching for an associated setup task in the Setup and Maintenance work area. Lookup types are grouped by task or task list. Each task or task list provides access only to certain lookup types. However, the generic tasks provide access to all the lookups types of a kind, such as all common lookups that are associated with the Manage Common Lookups task. Lookups defined for a specific application are managed using a task or task list associated with that application. Here are some of the common Service lookup tasks or task lists:

- Define Service Request Lookups
- Manage Contact Lookups
- Manage Customer Center Lookups
- Manage Service Request Severity
- Manage Service Request Channel Types

**Related Topics**

- How can I access predefined lookups?
Modifying Service Request Lookups: Explained

Administrators can modify lookups for service requests. Optionally, you can map status values to status types. By default, the following five Status types exist for service requests:

- New
- In Progress
- Resolved
- Waiting
- Closed

A service request always has one of these status types. However, administrators might want to display different labels for status types or change the display sequence. For example, this might be useful in situations where you want to distinguish between statuses such as "In Progress - Troubleshooting" versus "In Progress - Repairing." This procedure maps one or more statuses to status types.

The following tasks are used to modify the service request lookups:

- Manage Service Request Internal Priorities
- Manage Service Request Sources
- Manage Service Request Resolutions
- Manage Service Request Severities
- Manage Service Request Channel Types
- Manage Service Request Problem Types
- Manage Service Request Contact Relationship Types
- Manage Service Request Status Values

To modify service request lookups:

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Service Request functional area.
5. Select to display all tasks.
6. Click the task that you want to modify.
7. In Lookup Codes, click the lookup code that you want to modify.
8. Modify the fields to correspond to your needs.
9. Click Save and Close.
10. Repeat the procedure for all of the service request lookup tasks.

The service request lookup is modified.

FAQs for Lookups and Lookup Types
How can I edit lookups?

On any of the Manage Lookups pages, you can edit the existing lookup codes of a lookup type or add new lookup codes. To open the page, navigate to the Setup and Maintenance work area, open the panel tab and click Search to search for any of the following tasks:

- Manage Standard Lookups
- Manage Common Lookups
- Manage Set-enabled Lookups

Each task contains a predefined set of lookup types that are classified and stored. Open a task to search and edit the required lookup. However, you may not be able to edit a lookup if its configuration level doesn’t support editing.

Why can't I see my lookup types?

Lookup types are classified using tasks that involve a group of related lookups, such as Manage Geography Lookups. Each task gives you access only to certain lookup types. However, the generic tasks provide access to all lookup types of a kind, such as common lookups associated with the Manage Common Lookups task.

If the lookup types in an application are available in the standard, common, or set-enabled lookups view, they’re are central to an application. However, lookup types defined for a specific application are managed using the task or task list for that application.

Scheduled Processes: Explained

Run scheduled processes to manipulate a set of records for a specific business need, or to get printable output with information about certain records. Some processes do both, for example, to import records and provide a report about them.

Report Output

A scheduled process that provides output, or the output itself, is also referred to as a report.

- Many types of reports are available, for example regulatory statements or listings of records that meet specified parameters.
- Predefined templates determine the report layout.

Parameters

A scheduled process might have parameters that you can set to control which records are included or how they’re affected. For example, a process updates only the records that are effective within the date range that you define.

Submission

Each scheduled process that you run is based on a job. The job is the executable that determines what the process can do and what options you can set for the process.

You can submit the same process using different parameters and other settings. Each process submission has a unique process ID.
Process Sets
A process set is a scheduled process that's based on a job set, which contains multiple jobs for one process submission.

**Note:** In some cases, when you submit a scheduled process, the job logic causes other processes to automatically run. This isn’t the same as a process set.

**Related Topics**
- Process Sets: Explained
- Submitting Scheduled Processes and Process Sets: Procedure
- Managing Scheduled Processes That You Submitted: Points to Consider
- Creating Job Sets: Procedure

Profile Options and Scheduled Processes for SR Management

Profile Options for SR Management
Profile options let you configure and control application data centrally. Administrators and setup users manage profile options in the Setup and Maintenance work area. You can set various profile options and schedule job processes for service request management. Some of the profile options must be used along with job processes to achieve the results you want. For example, after setting the profile value for closing a resolved SR after N number of days, schedule a job process that closes SRs.

The following table lists the various profile options for service request management and their purposes.

<table>
<thead>
<tr>
<th>Profile Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_ENABLE_AUDIT_IN_SR</td>
<td>Enables or disables auditing of field value changes.</td>
</tr>
<tr>
<td>SVC_ASSIGN_TO_QUEUE_ON_CREATE</td>
<td>Assigns an SR to a queue automatically, when the SR is created.</td>
</tr>
<tr>
<td>SVC_SR_IN_RESOLVED_DAYS</td>
<td>Defines the number of days after which any resolved SR is automatically closed. This profile option must be used along with the Auto-Close Service Requests job process. For more information, see Scheduled Processes for SR Management.</td>
</tr>
</tbody>
</table>

Make sure the following two conditions are met:
- The SVC_SR_IN_RESOLVED_DAYS profile value must be set to 1 or greater for the Auto-Close Service Request job to run.
- The SVC_SR_IN_RESOLVED_DAYS profile value must be set to 0 to disable the Auto-Close Service Request job.
<table>
<thead>
<tr>
<th>Profile Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| SVC_SR_IN_WAITING_DAYS               | Defines number of days for which a service request must be in Waiting status before the SR is auto resolved. This profile option must be used along with the Auto-Close Service Requests job process. For more information, see Scheduled Processes for SR Management. Make sure the following two conditions are met:  
  - The SVC_SR_IN_WAITING_DAYS profile value must be set to 1 or greater for the Auto-Close Service Request job to run.  
  - The SVC_SR_IN_WAITING_DAYS profile value must be set to 0 to disable the Auto-Close Service Request job.                                                                                                                                                                        |
| SVC.AUTO_CLOSED_STATUS_CD            | Defines the status code to use for auto-closing service requests that have been in Resolved status for at least the number of days specified by SVC_SR_IN_RESOLVED_DAYS. This profile option must be used along with the Auto-Close Service Requests job process. For more information, see Scheduled Processes for SR Management.                                                                                                      |
| SVC.AUTO_RESOLVED_STATUS_CD          | Defines the status code to use for auto-resolving service requests that have been in Waiting status for at least the number of days specified by SVC_SR_IN_WAITING_DAYS. This profile option must be used along with the Auto-Close Service Requests job process. For more information, see Scheduled Processes for SR Management.                                                                                      |
| SVC.PUID_FORMAT                      | Defines the format for the unique reference number on each SR.                                                                                                                                                                                                                                                                             |
| SVC.PUID_PREFIX                      | Defines an optional prefix that can be included in an SR reference number.                                                                                                                                                                                                           |
| SVC_SR_DEFAULT_SEVERITY_CD           | Sets the default Severity value for a new SR.                                                                                                                                                                                                                                    |
| SVC_SR_DEFAULT_STATUS_CD             | Sets the default Status code for a new SR.                                                                                                                                                                                                                                       |
| SVC_SR_IN_DELETED_DAYS               | Defines the number of days after which a deleted SR is purged. An SR that is soft deleted can be retrieved. However, once purged, an SR can’t be recovered. This profile option must be used along with the Purge Deleted Service Requests job process to purge SRs. For more information, see Scheduled Processes for SR Management.                                                                 |
| SVC_ATTACHMENT_UI                    | Defines the attachment view for an SR. There are two types of attachment options you can provide:  
  - **Classic**: Enables users to select the type of attachment, category, and enter the name and description for each attachment.  
  - **Basic**: Enables users to only select file type, browse and upload attachments to the SR.                                                                                                                                                                   |
| SVC_EVENT_HISTORY_DAYS_TO_KEEP       | Specifies the number of days for which the SR event history details such as update and create must be retained. The data beyond this value is purged. Do not modify this value.                                                                                                                   |

### Scheduled Processes for SR Management

You can set various profile options and schedule job processes for service request management. Some of the profile options must be used along with job processes to achieve the results you want. For example, after setting the profile value for closing a resolved SR after N number of days, schedule a job process that closes SRs. Run scheduled processes to manipulate a set
of records for a specific business need, or to get printable output with information about certain records. Some processes do both, for example, to import records and provide a report about them.

The following table lists job processes that you can schedule to manage service requests.

<table>
<thead>
<tr>
<th>Job Process Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-Close Service Requests</td>
<td>Closes service requests that were resolved a certain number of days ago, and resolves service requests that have been in waiting status for a certain number of days. The number of days for both the scenarios is set using the SVC_SR_IN_RESOLVED_DAYS and SVC_SR_IN_WAITING_DAYS profile options respectively.</td>
</tr>
<tr>
<td>Purge Deleted Service Requests</td>
<td>Purges service requests and their child records that were deleted a certain number of days ago. The number of days is set using the SVC_SR_IN_DELETED_DAYS profile option.</td>
</tr>
<tr>
<td>Load and Update Cloud Metrics for Service</td>
<td>Performs incremental loading and updating of usage and business metrics that are targeted to cloud usage patterns.</td>
</tr>
<tr>
<td>Service Request Queue Assignment</td>
<td>Assigns queues to service requests. This job takes the following parameters.</td>
</tr>
<tr>
<td></td>
<td>• Work Object Code: Indicates business objects that get assigned to agents, such as, service requests. Expected Value: ORA_Service_Request_Work_Object</td>
</tr>
<tr>
<td></td>
<td>• Candidate Object Code: Indicates objects that are the possible pool of assignment candidates, such as queues. Expected Value: ORA_Queue_Candidate_Object</td>
</tr>
<tr>
<td></td>
<td>• Assignment Mode (List of Values: Classification, Matching, Scoring, Territory): Indicates the type of assignment processing. Matching is the only mode that is supported.</td>
</tr>
<tr>
<td></td>
<td>• View Criteria Name: Indicates the view criteria used to identify the service requests to be assigned. Expected Value: OpenSRsUnassignedToQueueByStripeCd</td>
</tr>
<tr>
<td></td>
<td>• Bind Variable: Indicates the bind variables required for the view criteria. Expected Value: BindStripeCd=ORA_SVC_CRM</td>
</tr>
<tr>
<td></td>
<td>• Metrics Logging Interval (default value is 100): Indicates the number of work objects in a subprocess before logging assignment metrics, such as update metrics after processing 100 SRs. This is used if your object support Enterprise logging for assignment.</td>
</tr>
<tr>
<td></td>
<td>• Diagnostic Mode (check box): Indicates if the process must be run in diagnostic mode to view the details of assignment processing in an output log.</td>
</tr>
<tr>
<td>Service Configuration Setup</td>
<td>Schedules service setup jobs, such as metrics. This job must be run once during implementation. Ensure that you run this job before you create service requests.</td>
</tr>
<tr>
<td>Aggregate Service Requests</td>
<td>Enables querying service request data for reporting, using the CRM - CRM Service Request Summary subject area. The recommended frequency for running this job is one hour.</td>
</tr>
<tr>
<td>Monitor Service Request Milestones</td>
<td>Ensures that the service request and milestone status are up-to-date and sends an email notification if compliance issues or warning flags are found.</td>
</tr>
<tr>
<td>Purge Service Event History</td>
<td>Evaluates the processed records and retains the data for the days specified in the profile option SVC_EVENT_HISTORY_DAYS_TO_KEEP while it purges the rest. The recommended frequency for running this job is daily.</td>
</tr>
</tbody>
</table>
To configure a scheduled process, do the following.

1. Sign in to Engagement Cloud as an administrator.
2. From the Navigator, select Scheduled Processes.
4. In the Schedule New Process dialog box, select Job as the Type option.
5. In the Name drop-down list, click Search to search and select the process that you want to configure.
6. In the Process Details dialog box for the selected job, click Advanced.
7. On the Schedule tab, select Using a schedule as the Run option.
8. Specify the Frequency for the job.
9. Select the Start Date and End Date for the job.
10. Click Submit.

>Note: The scheduled process is visible only to the user who creates the job.

<table>
<thead>
<tr>
<th>Job Process Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor Action Plan Actions</td>
<td>Evaluates the status of an action plan.</td>
</tr>
</tbody>
</table>
4 Setting Up Users and Security

Defining Setup Users: Overview

Among the initial activities when setting up Oracle Sales Cloud is the creation of users who perform setup tasks. Oracle creates an initial user for you when your Oracle Sales Cloud environment is provisioned. This initial user is configured to perform security tasks, which include the creation of other users and the granting of additional privileges. The initial user can create other users, known as setup users, to help with application setup. The setup user performs the tasks in Oracle Sales Cloud implementation projects, sets up enterprise structures, creates application users, and administers security.

Use the Manage Users task in the Setup and Maintenance work area to create setup users. For information about creating setup users, see Oracle Sales Cloud Getting Started with Your Sales Implementation guide.

Related Topics
- Oracle Sales Cloud Getting Started with Your Sales Implementation guide
- Oracle Sales Cloud Securing Oracle Sales guide

Setting Up Users and Security: Overview

Access to Oracle Sales Cloud functionality and data is secured using role-based access control. In a role-based access control model, users are assigned roles, and roles are assigned access privileges to protected system resources. Initial access to Oracle Sales Cloud is limited to one initial user that Oracle creates. Using this initial user, you create other required users, such as setup users, the sales administration user, and application users. You then provision each user with roles, which provide access to application functions and data.

Sales users who access the transactional UI, such as the Leads and Opportunities work areas, are created as resources and are known as sales resources.

To set up default preferences for users and roles, you access the Security Console as a setup user or other user with the IT Security Manager job role. Only setup users, or other users with the IT Security Manager job role, can access the Security Console. You perform user-related tasks both during implementation and later as requirements emerge. If you are a new customer, follow the steps in the Oracle Sales Cloud Getting Started with Your Sales Implementation guide. For ongoing maintenance of users, use the Users and Security functional area in Setup and Maintenance and the Users, Roles and Delegation task in the Navigator. For more information about creating and importing users, see the Oracle Sales Cloud Getting Started with Your Sales Implementation guide. For more information about setting up security and provisioning roles to users, see the Oracle Sales Cloud Securing Sales guide. You can find these guides on the Oracle Cloud Documentation site, linked in the Related Topics section of this topic.

LDAP Identity Store

The Oracle Cloud authentication providers access the LDAP identity store, which is a logical repository of enterprise user identity data. Your LDAP directory stores definitions of LDAP user accounts. In general, changes you make to user accounts are automatically synchronized between Oracle Sales Cloud and your LDAP directory server. However, you must also run
processes on a daily basis to manage information exchange between your application and your LDAP directory server. For information, see the chapter about setting up application security in the Securing Sales guide.

Setup Tasks in the UI and Other Setup Options

As a setup user, you access multiple tasks in Setup and Maintenance to create and maintain users. You also have additional setup options to consider. The following table describes these tasks and setup options.

<table>
<thead>
<tr>
<th>Setup Task or Option and Navigation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Job Roles Task:</td>
<td>Oracle provides many predefined job roles. The relevant Sales Cloud roles are listed in the Getting Started with Your Sales Implementation guide.</td>
</tr>
<tr>
<td>Navigation: Setup and Maintenance &gt; Sales Offering &gt; Users and Security functional area</td>
<td>You perform the Manage Job Roles task to:</td>
</tr>
<tr>
<td></td>
<td>• Review the role hierarchy of a job or abstract role.</td>
</tr>
<tr>
<td></td>
<td>• Create custom job and abstract roles.</td>
</tr>
<tr>
<td></td>
<td>• View the roles assigned to a user and list the users who have a specific role.</td>
</tr>
<tr>
<td></td>
<td>This task opens the Roles tab of the Security Console.</td>
</tr>
<tr>
<td>Manage Duties Task:</td>
<td>You perform the Manage Duties task to:</td>
</tr>
<tr>
<td>Navigation: Setup and Maintenance &gt; Sales Offering &gt; Users and Security functional area</td>
<td>• Review the duties of a job or abstract role.</td>
</tr>
<tr>
<td></td>
<td>• Manage the duties of a custom job or abstract role.</td>
</tr>
<tr>
<td></td>
<td>• Create custom duty roles.</td>
</tr>
<tr>
<td></td>
<td>This task opens the Roles tab of the Security Console.</td>
</tr>
<tr>
<td>Manage Data Security Policies Task</td>
<td>You use the Manage Data Security Policies task to manage the data security policies that determine grants of entitlement to a user or role on an object or attribute group. This task opens the Roles tab of the Security Console.</td>
</tr>
<tr>
<td>Navigation: Setup and Maintenance &gt; Sales Offering &gt; Users and Security functional area</td>
<td></td>
</tr>
<tr>
<td>Users, Roles and Delegations Task</td>
<td>You create application users in the UI using the Users, Roles and Delegations task. A user with the IT Security Manager job role performs the Manage Users tasks.</td>
</tr>
<tr>
<td>Navigation: Navigator &gt; Users, Roles and Delegations item or Setup and Maintenance &gt; Sales Offering &gt; Users and Security functional area</td>
<td>Note: You can’t perform bulk imports of data into Sales Cloud using the Import Worker Users task available from the Users and Security functional area task list. However, you can create users by importing legacy users from a file using the Manage File Import Activity task available from the Setup and Maintenance work area. For information on importing users, see the Getting Started with Your Sales Implementation guide.</td>
</tr>
<tr>
<td>Manage HCM Role Provisioning Rules Task</td>
<td>Oracle provides predefined role mapping rules for provisioning many of the standard job roles included with the application. However you can create any additional role mappings you need to control the provisioning of roles to application users using the Manage HCM Role Provisioning Rules task. For example, you can create a role mapping to provision the Channel Sales Manager role automatically to specified sales managers.</td>
</tr>
<tr>
<td>Navigation: Setup and Maintenance Sales OfferingUsers and Security functional area</td>
<td></td>
</tr>
</tbody>
</table>
### Setup Task or Option and Navigation

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>File-Based Data Import</td>
<td>You can import users in bulk using file-based data import. See the Getting Started with Your Sales Implementation guide for more information.</td>
</tr>
<tr>
<td>Import Partner Users Task</td>
<td>You can also import partner contact data using the Import Partner Users task. For more information, see the Oracle Sales Cloud Getting Started with Your Partner Relationship Management Implementation guide.</td>
</tr>
<tr>
<td>Single Sign-On Authentication</td>
<td>Single sign-on authentication, which enables users to sign in once and access multiple applications, is optionally available for Oracle Sales Cloud user authentication. If your enterprise has moved from a traditional on-premises environment to an Oracle Cloud implementation, you might want to use your existing identity management solution for authenticating your employees in Sales Cloud, and might also want to provide a single sign-on experience. Implementing federated single sign-on lets you provide users with single sign-on access to applications and systems located across organizational boundaries. For additional information, see Oracle Applications Cloud Service Entitlements (Doc ID 2004494.1) on My Oracle Support at <a href="https://support.oracle.com">https://support.oracle.com</a>.</td>
</tr>
<tr>
<td>Resetting User Passwords</td>
<td>Setup users, who are provisioned with the IT Security Manager job role, can use the Users tab in the Security Console work area to reset passwords for all application users. Users who can’t access the Security Console can reset only their own passwords using the Set Preferences link in the Settings and Actions menu available by clicking their user name in the application or by using the Forgot Password link on the sign-in page. See the Getting Started with Your Sales Implementation guide for more information.</td>
</tr>
<tr>
<td>Updating Email Addresses</td>
<td>Use the Users tab in the Security Console work area to change user email addresses. You can use the procedure described in this topic to update addresses of both setup users and sales users. If you are updating the email addresses of sales users, then you can also use the same import process you use to create them. See the Getting Started with Your Sales Implementation guide for more information.</td>
</tr>
</tbody>
</table>

**Note:** Other data security tasks listed in the Users and Security functional area task list do not apply to Oracle Sales Cloud. Follow the guidance in the Getting Started with Your Sales Implementation guide and the Securing Sales guide.

### About Sales Resources

After creating your setup users and the sales administrator, you create sales resources. Creating resources is covered in the Getting Started with Your Sales Implementation guide. Ongoing maintenance of sales resources is performed by the sales administrator and by other transactional users, such as sales managers. Sales resources themselves also can update their own information. Some of these tasks (setup-related) are covered in this chapter. For tasks related to maintaining sales resource information, such as profiles, photos, and the like, refer to the Oracle Sales Cloud Using Sales guide.

#### Related Topics
- Oracle Sales Cloud Getting Started with Your Sales Implementation guide
- Oracle Sales Cloud Securing Sales guide
- Authentication chapter of the Oracle Sales Cloud Securing Sales guide
- Managing Resources chapter of the Oracle Sales Cloud Using Sales guide
About Resources and Resource Management: Explained

The following topic explains resources and the different ways you can logically group them. See the related links for more information on each.

Resource
A resource is an application user who participates in business processes, such as Customer Service Manager or Help Desk Agent. You must identify or import resources before you can associate them with resource organizations or work objects.

Resource Role
You use resource roles to define role provisioning rules. For example, using a provisioning rule, you can assign the Customer Service Manager job role to a user with the Customer Service Manager resource role. Oracle Applications Cloud includes some predefined resource roles. You may either use these predefined, out of the box resource roles, or you can create your own resource roles.

Resource Directory
The Resource Directory offers detailed information about all the resources within a deploying organization. The Resource Directory also enables you to find and communicate with other resources, and to network and collaborate with them. You can access the Resource Directory using the following path: Navigator > Resource Directory. You can also perform some of the functions of the resource directory using the Manage Resources setup task.

Resource Organization
You create resources and provision the permissions that the resources need to do their jobs. In the process, you also build the organization chart of your organization. You can assign organization usage information to resource organizations to classify them based on how on how you want to use them. For instance, you can assign resource organizations engaged in customer service activities to the Customer Service Organization usage. This allows you to sort organizations based on their usage.

A resource organization becomes a primary resource organization by usage for a resource, if you meet the following criteria:

- The resource must be a member of the concerned organization.
- The resource organization must be classified as an organization with the specific usage.

For example, if you classify the resource organization as a customer service organization by usage, then the resource organization becomes the primary resource organization for the resource for customer service.

Resource Team
A resource team is a temporary group of resources formed to complete a business task. A resource team can comprise resource organizations, resources, or both. A resource team is neither hierarchically structured nor intended to implement an organization structure. You can use resource teams as a quick reference to groups of related resources to which you can quickly assign work objects.
Related Topics

- Resource Role Assignment: Explained
- Resource Organizations and Organization Usage: Explained

Resource Team Information

Managing Resource Teams: Procedure

This procedure describes how to manage resource teams. A resource team is a temporary group of resources formed to complete a business task. A resource team cannot be hierarchically structured and is not intended to implement an organization.

Creating Resource Teams

To create resource teams:

2. Click the Create action menu option or button.
   The Create Team page appears.
3. Enter an appropriate team name.
4. Optionally, enter a team description and specify team usage, resource members, and organization members.
5. Click Save and Close.

Editing Resource Teams

To edit resource teams:

2. Search for the resource team that you want to edit.
   You can search by entering criteria such as the team name, number, and usage. You can also use the saved searches.
3. Select the resource team you want to edit from the Search Results region and click its name to navigate to the Edit Team page.
4. On the Edit Team page, you can edit the team’s details such as the team name, description, usage, resource members and organization members.
5. Click Save and Close.

Resources: How They Work Within a Team

This topic explains how resources work within a team.

You can include resources from different resource organizations to work together on a work object as members of the same resource team. You can also include the entire resource organization into a resource team. The resource organization
membership and their hierarchy determine what a resource can do. Resource teams provide a flexible way of bringing resources together without any organizational or hierarchy-based restrictions.

Assigning Resources to Teams
You can assign identified resources to teams and assign them roles within the team. Each resource can have a specific role within a team. A resource may play different roles in different teams.

Resource Team Membership and Role Assignment Components: How They Work Together
This topic explains the team membership and role assignment for resources.
Resources who are team members can be assigned different roles within the team. These roles do not necessarily reflect the roles these resources might play in resource organizations. Depending on the task requirements of the team, roles are assigned to resources. Based on the role assignment, resources can access data related to the tasks in the team. You can manage the resources in a team using the Manage Resource Teams task. You can access the Manage Resource Teams task as follows: Click Navigator > Resource Directory > Tasks > Manage Resource Teams.

Resource Team Membership
A resource can belong to multiple teams depending on the requirements of these teams and the skills that the resource offers. This does not affect the resource's membership with organizations within the deploying company.

Role Assignment
Resources have specific roles to play in the team to which they belong. Each of these roles can be different. Also, these roles can be different from the roles assigned to the same resources in resource organizations. Thus, a resource can be a manager in one team and a member in another simultaneously.

FAQs for Define Resource Team Information
What's a resource team?
A resource team is a group of resources formed to work on work objects. A resource team can comprise resource organizations, resources, or both. A resource team is neither hierarchically structured nor intended to implement an organization structure. You can use resource teams as a quick reference to groups of related resources to which you can quickly assign work objects.

Note: You can either individually assign the members of a team to a task or assign entire teams to tasks.
What's the difference between a resource organization and a resource team?

A resource organization is an organization whose members are resources. Resource organizations are used to implement sales organizations, partner organizations, and so on.

A resource team is a temporary group of resources formed to work on work objects. A resource team may contain a resource organization or resources or both. A resource team cannot be hierarchically structured and is not intended to implement an organization.

Can I assign multiple resource roles to a team member at the same time?

Yes. Resources within resource teams can have multiple resource roles. You can add roles to a resource in a resource team using the Manage Resource Teams task. Select the resource and click the Edit button to assign additional roles to the resource.

About Security Roles: Explained

Many job roles and duty roles are predefined in the Service offering. The following table lists the main predefined job roles specific to this product area.

<table>
<thead>
<tr>
<th>Job Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Service Manager</td>
<td>Manages a group of Customer Service Representatives. The person in this job role manages the queues of open service requests. Also this person ensures that work is being assigned and resolved by the pool of resources. This individual also monitors incoming service requests.</td>
</tr>
<tr>
<td>Customer Service Representative</td>
<td>Responds to a variety of customer requests. The person in this job might receive requests to help with a product or service problem or to clarify a pricing or ordering question. This individual is usually assigned to one or more queues of service requests that are used to categorize incoming service requests by customer type, category, or product.</td>
</tr>
<tr>
<td>Knowledge Analyst</td>
<td>Creates, curates, and maintains knowledge articles. The person in this job supports the work of Customer Service Managers and Customer Service Representatives as they manage customer issues.</td>
</tr>
<tr>
<td>Knowledge Manager</td>
<td>Manages a knowledge instance. The person in this job manages the administration and operations of a knowledge program. In addition, this person ensures that knowledge can be created and maintained by analysts and found by knowledge users.</td>
</tr>
</tbody>
</table>

In addition to the job roles, the following table lists the duty roles specific to the Service offering.
### Duty Role Description

<table>
<thead>
<tr>
<th>Duty Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Request Troubleshooter</td>
<td>Manages all aspects of the Service Request object. A customer service representative is typically given this duty role.</td>
</tr>
<tr>
<td>Service Request Contributor</td>
<td>Creates or adds information to service requests. This person has basic Service Request Management capabilities. This person can't respond to customers, assign service requests to a user, or delete attachments from service requests. This role is typically given to non-service users, for example, sales representatives, product managers and engineers.</td>
</tr>
<tr>
<td>Service Request Power User</td>
<td>Do all the tasks that a troubleshooter can, with the added privileges to manage queues. Customer service managers are typically given this duty role.</td>
</tr>
<tr>
<td>Service Request Administrator</td>
<td>Do all the tasks that a power user can do. Additionally, this person can administer all aspects of the application, including the setup. CRM application administrators are typically given this role.</td>
</tr>
<tr>
<td>Service Request Channel User</td>
<td>Manages all aspects of a service request that is associated with a partner account. This duty is similar to the service request troubleshooter.</td>
</tr>
<tr>
<td>Service Request Partner User</td>
<td>Submits service requests to get assistance with partner sales or support issues. This duty is granted to a contact from a partner account.</td>
</tr>
</tbody>
</table>

### Role Provisioning Rules for Service Resource Roles

The following resource roles are predefined for the Service offering: service manager, and service representative. You must create role provisioning rules for these resource roles to automatically assign new service users with the job roles and abstract roles required to do their job. For additional information about creating these rules, review the related topic about Creating Rules to Automatically Provision Job Roles to Sales Users.

Use the following table as a guideline to create the role provisioning rules for Service offering:

<table>
<thead>
<tr>
<th>Provisioning Rule Name</th>
<th>Condition</th>
<th>Abstract Roles Provisioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Manager</td>
<td>HR Assignment is Active</td>
<td>Customer Service Manager</td>
</tr>
<tr>
<td></td>
<td>Resource Role is Service Manager</td>
<td>Resource</td>
</tr>
<tr>
<td>Service Representative</td>
<td>HR Assignment is Active</td>
<td>Customer Service Representative</td>
</tr>
<tr>
<td></td>
<td>Resource Role is Service Representative</td>
<td>Resource</td>
</tr>
</tbody>
</table>

### Related Topics

- Role-Based Access Control: Explained
- Creating Rules to Automatically Provision Job Roles to Sales Users
Setting Up Service Request Visibility Based on Queue

When users view lists of service requests or create user-defined searches, their access is based on the cumulative set of data security policies assigned to all the roles associated with them. However, you can restrict their access based on their queue membership.

To ensure that the users see only the SRs that are in their queue, you must complete the following processes in the Security Console as described later in this topic:

- If the users have been assigned other data security policies that grant them access to a larger set of SRs, then remove such data security policies from the users.
- Grant queue-based visibility to service requests for specific roles. Consequently, users with these specific roles can see only the service requests assigned to the queues where they’re a resource member.

With this new data security policy, your company has the additional option to ensure that all predefined and user-defined searches are limited only to queue membership for a set of users.

Removing Data Security Policies from Users

If your users have existing service duty roles and you don’t want them to see all the service requests that they can currently see, you must remove those data security policies from the users.

To remove the data security policies from your users:

1. Copy the predefined duty roles given to your users.
   
   For more information about copying roles, see "Copying and Editing Duty Roles" in the Oracle Sales Cloud Securing Sales guide at the following location: https://docs.oracle.com/en/cloud/saas/sales/18b/oscus/creating-job-abstract-and-duty-roles.html#OSCUS1717200

2. Remove the data security policies that you do not want these users to have.
   
   For more information about removing or creating data security policies, see "Managing Data Security Policies: Explained" in the Oracle Sales Cloud Securing Sales guide at the following location: https://docs.oracle.com/en/cloud/saas/sales/18b/oscus/creating-job-abstract-and-duty-roles.html#OSCUS1717419

   📌 Note: Complete the following procedure to assign only the queue-based data security policies to the users.

Assigning Data Security Policies Based on Queue

The predefined roles don’t have queue-based service request visibility.

To assign the queue-based data security policies to your users:

1. In the Security Console, create a new job role and click Next.
   
   For more information about creating or editing a job role, see "Creating Job or Abstract Roles" in the Oracle Sales Cloud Securing Sales guide at the following location: https://docs.oracle.com/en/cloud/saas/sales/18b/oscus/creating-job-abstract-and-duty-roles.html#OSCUS2102030
2. For the new job role created in the previous step, create a new data security policy by specifying the following attributes:
   o Give an appropriate name to the policy.
   o Select Data Resource = Service Request Header.
   o Select Data Set = Select by instance set.
   o Select Data Condition Name = Access the service request header for table SVC_SERVICE_REQUESTS where the customer relationship management service request is assigned to a queue that they're a member of.
   o Select the appropriate actions. Typically, you must select all actions except Delete.
3. Click OK and continue to the Users section of the process for creating a job role.
4. Add the users to whom you want to assign this data security policy.
5. Save the job role.

Related Topics
• Managing Data Security Policies: Explained
• Creating Job or Abstract Roles: Procedure
• Copying and Editing Duty Roles: Procedure
• Copying Job or Abstract Roles: Procedure

Setting Up Service Request Visibility Based on BU

When users view lists of service requests or create user-defined searches, their access is based on the cumulative set of data security policies assigned to all the roles associated with them. However, you can restrict their access based on their Business Unit (BU) membership.

To ensure that the users see only the service requests that are assigned to the business units where they’re a resource member, you must complete the following processes in the Security Console as described later in this topic:
   • If the users have been assigned other data security policies that grant them access to a larger set of service requests, then remove such data security policies from the users.
   • Grant BU-based visibility to service requests for specific roles. Consequently, users with these specific roles can see only the service requests assigned to the business units where they’re a resource member.

With this new data security policy, your company has the additional option to ensure that all predefined and user-defined searches are limited only to business unit membership for a set of users.

Removing Data Security Policies from Users

If your users have existing service duty roles and you don’t want them to see all the service requests that they can currently see, you must remove those data security policies from the users.

To remove the data security policies from your users:
   1. Copy the predefined duty roles given to your users.
For more information about copying roles, see "Copying and Editing Duty Roles" in the Oracle Sales Cloud Securing Sales guide at the following location: https://docs.oracle.com/en/cloud/saas/sales/18b/oscus/creating-job-abstract-and-duty-roles.html#OSCUS1717200

2. Remove the data security policies that you do not want these users to have.

For more information about removing or creating data security policies, see "Managing Data Security Policies: Explained" in the Oracle Sales Cloud Securing Sales guide at the following location: https://docs.oracle.com/en/cloud/saas/sales/18b/oscus/creating-job-abstract-and-duty-roles.html#OSCUS1717419

Note: Complete the following procedure to assign only the BU-based data security policies to the users.

Assigning Data Security Policies Based on BU

The predefined roles don't have service request visibility based on business unit.

To assign the data security policies based on business unit to your users:

1. In the Security Console, create a new job role and click Next.

   For more information about creating or editing a job role, see "Creating Job or Abstract Roles" in the Oracle Sales Cloud Securing Sales guide at the following location: https://docs.oracle.com/en/cloud/saas/sales/18b/oscus/creating-job-abstract-and-duty-roles.html#OSCUS2102030

2. For the new job role created in the previous step, create a new data security policy by specifying the following attributes:
   o Give an appropriate name to the policy.
   o Select Data Resource = Service Request Header.
   o Select Data Set = Select by instance set.
   o Select one of the following data condition names:
     - Access the service request header for table SVC_SERVICE_REQUESTS where the customer relationship management service request is associated with my business units or the application default business unit.
     - Access the service request header for table SVC_SERVICE_REQUESTS where the partner service request is associated with my business units or the application default business unit.

3. Select the appropriate actions. Typically, you must select all actions except Delete.
4. Click OK and continue to the Users section of the process for creating a job role.
5. Add the users to whom you want to assign this data security policy.
6. Save the job role.

Related Topics

- Managing Data Security Policies: Explained
- Creating Job or Abstract Roles: Procedure
- Copying and Editing Duty Roles: Procedure
- Copying Job or Abstract Roles: Procedure
5 Managing Catalogs

Defining a Catalog for the Service Offering: Explained

In most implementations, you must create a catalog of products and services. If you have already defined a catalog with the Base usage for Sales, it can be used in the Service offering. Alternatively, you have the option to create and manage a distinct catalog for the Service offering.

Before deciding whether to create a distinct catalog for the Service offering, consider that using the same catalog as Sales means you have the same hierarchy of product groups used to categorize service requests. For some implementations, you can use the same catalogs. However, in other implementations, the Sales catalog might not have the right granularity to properly categorize service issues. In these situations it would be appropriate to create a separate hierarchy of product groups and products, specifically for the Service offering.

Using an Existing Sales Catalog for the Service Offering

To use the same catalog for your Service offering that you use for your Sales implementation, you must add the root product group for the Sales catalog in the Service offering.

To use an existing Sales catalog in your Service offering:

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Sales.
4. Select the Sales Catalog functional area.
5. Select the Manage Product Group Usage task.

   The Manage Product Group Usage page appears.
6. Click the Service usage.
7. Under the Service: Details section, in the Product Groups tab, add the root product group for the sales catalog.
8. Save the record.

Defining a Distinct Catalog for the Service Offering

Defining a catalog for the Service offering is the same procedure that you use for Sales. The only difference is that for Product Group Usage, Sales uses the Base usage, whereas Service uses Service usage. Use the following procedure to begin defining a catalog for the Service offering. Use the related Sales documentation to complete this procedure.

To begin defining a distinct catalog for the Service offering:

1. In the Manage Product Group Usage page, click the Service usage.
2. Create your catalog of products and services for the Service offering.

Related Topics

- About the Sales Catalog
- Creating the Root Product Group
Sales Catalogs: Overview

Using sales catalogs in Oracle Sales Cloud lets you:

- Use product group as a territory dimension so that assignments can be made based on product.
- Give salespeople a mechanism to add product revenue to opportunities.
- Allow salespeople to add products to leads.
- Have product revenue available in forecasting and salesperson quota.

**Note:** While you can include individual products (also known as inventory items) in your catalog, they're not required unless you're integrating with a product application downstream, such as Oracle Configure, Price and Quote (CPQ) Cloud. For information on the setup of individual products, see the topics on Sales Cloud products.

To get started creating your sales catalog, see the topic, Creating the Sales Catalog: Getting Started.

Sales Catalog Key Features

The following are the main features of the sales catalog:

- Quickly build and deploy sales catalogs in a single administration UI.
- Catalog administration tool allows you to build product groups in a hierarchy.
- Product group display name and description can be translated into different languages.
- Use file-based import to import product groups rather than having to enter them in the UI.
- Use the sales products UI to create individual products that you then can add to the product group hierarchy.

Product Group Hierarchy Example

The following figure shows an example of a product group hierarchy.

In the example:

- At the top of the product group hierarchy is the root product group, named Special Deals.
- The nested groups begin with the child groups of the Special Deals root group. These include: Men, Women, and Kids.
- Within the Kids group, more nested groups appear, including Girls and Boys.
- Within the Girls group, a child group called Apparel appears.
- Within the Apparel group, further nesting occurs, with the groups Pants, T-shirts, and Dresses.
Together, the root group and configuration of parent and child groups make up the sample hierarchy.

![Diagram of product groups]

**Related Topics**
- Validating the Sales Catalog
- Running Refresh Denormalized Product Catalog Table Process
- Creating a Sales Catalog: Worked Example

**Creating the Sales Catalog: Getting Started**

Sales catalogs organize the products and services that you sell in a hierarchy of product groups. Your salespeople select product groups from the sales catalog when they create leads and opportunities, so you must create at least one sales catalog. You can also use the product groups as a dimension for defining sales territories and for preparing management reports.

**High-Level Setup Steps**

You must perform several steps to set up the sales catalog. The following table shows the high-level setup steps and where to find more information about the step.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Where to Find More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create the root product group.</td>
<td>Create the root product group. The root catalog or root product group is the top of the product group hierarchy. All other product groups are nested underneath.</td>
<td>Creating the Root Product Group section in this topic</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td>Where to Find More Information</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-------------------------------</td>
</tr>
</tbody>
</table>
|      | **Create the product group hierarchy.** | • Creating the Product Group Hierarchy section in this topic  
• The topic, Importing Products and Product Groups in the Oracle Sales Cloud Getting Started with Your Sales Implementation guide  
• The Importing Product Groups chapter of the Oracle Sales Cloud - Understanding File-Based Data Import and Export guide |
|      | Perform this step in the product groups pages in Setup and Maintenance. | |
|      | Publish the sales catalog. | • Publishing the Sales Catalog section in this topic  
• Running Refresh Denormalized Product Catalog Table topic |
|      | Perform this step in the product groups pages in Setup and Maintenance. | |
|      | When you publish a catalog, the scheduled process, Refresh Denormalized Product Catalog Table for BI, runs automatically to update the current view of the product group hierarchy in consuming applications. | |
|      | Set the catalog’s usage to Base. | • Enabling the Sales Catalog topic  
• Running Refresh Denormalized Product Catalog Table topic |
|      | Perform this step in the product groups pages in Setup and Maintenance. | |
|      | Each time you make a new assignment of Base to a root product group, you must run the scheduled process, Refresh Denormalized Product Catalog Table for BI. If you do not run the process, your product group hierarchy may not appear in the consuming applications. | |
|      | Set the browse catalog profile option. | Enabling the Sales Catalog topic |
|      | Set the profile option, Browse Sales Catalog in Opportunities Enabled, to Yes to enable Browse Sales Catalog button on the Products table in the simplified UI. Perform this step in the Manage Opportunity Profile Options task in Setup and Maintenance. | |
|      | If you have set up the browse catalog feature, configure search and browse options. | Setting Options for Sales Catalog Searching and Browsing topic |
|      | Verify your setups. | Validating the Sales Catalog topic |
## Implementing Service in Engagement Cloud

### Chapter 5

### Managing Catalogs

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Where to Find More Information</th>
</tr>
</thead>
</table>
| **Create products (items).** | Optionally, set up products to be able to use products in your sales catalog. You can use either Oracle Sales Cloud sales products UI or the product model, part of Oracle Supply Chain Management (SCM) Cloud.  
- If you do not have an integration with an order management application downstream, such as Oracle Configure, Price, and Quote (CPQ) Cloud, simply using product groups are sufficient for Oracle Sales Cloud and for integrations with other Oracle Applications Cloud services. | If using Oracle Sales Cloud products:  
- Setting Up Sales Products chapter in the Oracle Sales Cloud - Implementing Sales guide  
If using Oracle Supply Chain Management (SCM) Cloud product model:  
- Oracle SCM Cloud - Implementing Product Management  
- Oracle SCM Cloud - Using Product Master Data Management  
These guides are available on the Oracle Help Center. |
| **Add products to the catalog.** | Optionally, add the products you have created to the product group hierarchy. | Adding Products to the Catalog: Procedure topic |
| **Set up eligibility rules for products** | Optionally, implement eligibility rules that enable salespeople to check product eligibility in opportunities. | The topic, Sales Product Eligibility: Explained |
| **Set territory filtering options for the runtime UI.** | Optionally, configure whether the sales catalog displays only product groups and products within a user’s sales territories, or if it displays all product groups and products defined in the catalog. Configure whether to have territory filtering off by default, and whether to let users turn the territory filter on or off. | Filtering Catalog Display by Territories: Explained topic |
| **Integrate with Oracle Configure, Price, and Quote (CPQ) Cloud for additional capabilities.** | Optionally, use the prebuilt Sales Cloud and Oracle CPQ Cloud integration. This integration lets sales representatives manage quotes and orders from accounts and opportunities, finalize pricing and proposals in Oracle CPQ Cloud, update opportunity revenue with quote lines for accurate forecasting, and access proposal documents from within Sales Cloud. | The article CPQ Cloud to Oracle Sales Cloud Integration White Paper (Doc ID 2015009.1) available on My Oracle Support |

### Creating the Root Product Group

The root product group is the top-level product group in your catalog. The display name you use appears in the UI for users.

Use the following procedure to create the root product group.

1. Sign in as the sales administrator or as a setup user and navigate to the Setup and Maintenance work area.  
2. Navigate to the Sales offering icon and click the Setup button.  
   The Setup: Sales page appears.  
3. In the list of functional areas, click the Sales Catalog area.
A list of required tasks for the area appears.

4. In the list of tasks, click the **Manage Product Groups** task.

   The Manage Product Groups page appears.

5. Click the **Create** icon.

6. In the **Name** field, enter a unique name without spaces. This is the internal name of the group.

7. In the **Display** field, enter the product group display name. This is the name that displays in the UI to users.

8. Optionally, enter a description.

9. Optionally, enter the effective start and end dates.

10. Select the following check boxes:

    - **Active**: Only active product groups are available for use in the consuming applications.
    - **Root Catalog**: The root catalog is the top product group in the hierarchy. All other product groups created under it are considered subgroups. You can only add root catalogs to the Base usage in the Manage Product Group Usage page. Adding your catalog to the Base usage is a required step to enable the catalog for use in consuming applications.
    - **Locked**: This check box may already be checked. A product group must be "locked" to be edited.

11. Deselect the **Allow Duplicate Children** check box. This setting ensures that product groups and products do not appear multiple times in the hierarchy.

12. Optionally, deselect the **Allow Selection** check box. This setting ensures that product groups do not appear in the runtime UI.

13. Click **Save and Close**.

14. Verify that the root product group appears in the Manage Product Groups pane.

**Creating the Product Group Hierarchy**

If you are manually creating the product group hierarchy in the UI, create the remaining product groups under the root product, using the following steps:

1. Click the root product group in the side pane.

When viewing product groups in the Manage Product Groups page, you have two view options:

- **List view**: When you first enter the Manage Product Groups page, the product groups are shown as a list of folders. Click the tree view icon to enter tree view.

- **Tree view**: Tree view shows the product groups as nested parent-child groups. To return to list view, click the list view icon. In order to see the list view icon and the list of product groups, you may need to expand the Manage Product Groups pane.
2. In the Manage Product Groups page, click the **Subgroups** tab in the main work area.

   The product group information for the selected group appears in the main work area.

   🔄 **Tip:** A product group must be "locked" to be edited, so ensure that the parent of the product group you are creating is locked.

3. Click the **Create** icon.

4. In the Create Subgroup dialog box, enter the product group information.

   o In the **Name** field, enter a unique name without spaces.
   o In the **Display** field, enter the product group display name.
   o Optionally, enter a description.
   o Optionally, enter the effective start and end dates.
   o Select the following check boxes:

     • **Active:** Only active product groups are available for use in the consuming applications.
     • **Root Catalog:** Do not select the **Root Catalog** check box. You can have only one root catalog.

   o Deselect the **Allow Duplicate Children** check box. This ensures that product groups and products do not appear multiple times in the hierarchy.

5. Click **Save and Close**.

6. Verify that the product subgroup is visible in the Manage Product Groups pane. If the new subgroup does not appear, then click **View** and then **Refresh**.

7. Repeat the steps to create additional levels in your sales catalog hierarchy.
Publishing the Sales Catalog

After you create your product group hierarchy, use the following steps to publish your sales catalog. You must publish the root group at minimum, to be able to associate it to the Base usage. See the topic, Enabling the Sales Catalog, for more information.

1. Lock the root product group and the remaining groups in your hierarchy that you want to make available to end users.
2. Select the root group and click the Publish button.

⚠️ Caution: When you publish a node in the hierarchy, the application attempts to also publish all of the locked product groups. Therefore, if you have product groups in the application that you do not want published, be sure to unlock them so that they do not get published with the root and its subgroups.

3. Click Yes in the Confirm Publish dialog box.
4. Click OK on the confirmation message that is displayed.
5. Click Save and Close.

Related Topics

- Validating the Sales Catalog
- Running Refresh Denormalized Product Catalog Table Process
- Best Practices for Sales Catalog Setup

Adding Products to the Catalog: Procedure

After you have created products, you can add them to the product groups that make up the sales catalog hierarchy.

The source for your products can be either products created in the sales Products screens or in the Oracle Supply Chain Management (SCM) Cloud Products screens. For more information on product creation, see the topics about sales products.

Adding Products to the Catalog

Use the following procedure to add products to the sales catalog product group hierarchy.

1. Sign in as the sales administrator and navigate to the Setup and Maintenance work area.
   The Setup page appears with an offering selected.
2. In the Setup page, select the Sales offering.
   The Setup: Sales page appears with a list of functional areas.
3. In the list of functional areas, click the Sales Catalog and Products area.
   A list of required tasks for the area is displayed.
4. In the list of tasks, select the Manage Product Groups task.
   The Manage Product Groups page appears.
5. In the Manage Product Groups page, in the product group hierarchy, select the product group that you want to add products to.
6. Lock the product group for editing by clicking the **Lock** button.
7. Click the **Products** tab for the product group you selected.
8. In the **View** filter, ensure that the Administration view is selected.
9. In the products table, select **Actions > Select and Add**.

   The Select and Add: Products window appears.
10. Search for and select the product you're adding.
11. Click **Apply** and then **OK** in the select and add window.

   The application returns to the Manage Product Groups page with the product added to the product group.
12. Click the **Publish** button to publish the product group.
13. Click **Yes** in the Confirm Publish dialog window and then dismiss the confirmation message.

   The product group is automatically published.
14. Save your changes.

**Related Topics**
- Validating the Sales Catalog
- Running Refresh Denormalized Product Catalog Table Process
- Sales Products: Overview

---

**Enabling the Sales Catalog**

After you have created your product group hierarchy in Oracle Sales Cloud and optionally added individual products (items), you must enable the sales catalog for use in the consuming applications, such as opportunities and leads.

To enable the sales catalog, perform the following steps:

1. Set the Browse Sales Catalog profile option. This optional step enables the display of the browse button in the UI.
2. Associate the catalog with Base usage: This required step makes the catalog visible in the consuming applications.

**Set the Browse Sales Catalog Profile Option**

Set the profile option, Browse Sales Catalog in Opportunities Enabled to Yes to enable Browse Sales Catalog button on the opportunity Products table in the simplified UI. Use the following steps:

1. Sign in as the sales administrator and navigate to the Setup and Maintenance work area.

   The Setup page appears with an offering selected.
2. In the Setup page, select the **Sales** offering.

   The Setup: Sales page appears with a list of functional areas.
3. In the list of functional areas, click the **Opportunities** area.

   A list of required tasks for the area is displayed.
4. In the list of tasks, click the **Manage Opportunity Profile Options** task.

   The Manage Opportunity Profile Options page appears.
5. In the search region, enter `Browse Sales Catalog in Opportunities Enabled` in the **Profile Display Name** field.
6. Click **Search**.
7. In the list that is returned, click on the profile option name link.
8. Set the profile option value to **Y**.
9. Save your changes.

### Associate the Root Catalog with Base Usage

To enable a sales catalog for use in Oracle Sales Cloud, you associate it with a "usage" called the Base usage. Use the following steps:

1. Sign in as the sales administrator and navigate to the Setup and Maintenance work area.
   
   The Setup page appears with an offering selected.
2. In the Setup page, select the **Sales** offering.
   
   The Setup: Sales page appears with a list of functional areas.
3. In the list of functional areas, click the **Sales Catalog and Products** area.
   
   A list of required tasks for the area is displayed.
4. In the list of tasks, click the **Manage Product Group Usage** task.
   
   The Manage Product Group Usage page appears.
5. In the Manage Product Group Usage page, select the **Base** record.

   **Tip:** If a product group is already associated with the Base usage in the Details section in the portion of the screen, then you can remove the product group by selecting it and clicking the **Delete** icon.

6. In the Details section, click the **Select and Add** icon.
7. In the dialog box that appears, search for the root catalog that you just created.
8. Select the record and click **OK**.
9. In the Manage Product Group Usage page, click **Save and Close**.

   **Note:** Each time you make a new assignment of Base to a root product group, you must run the scheduled process, Refresh Denormalized Product Catalog Table for BI. If you do not run the process, your product group hierarchy may not appear in the consuming applications. See the topic, Running Refresh Denormalized Product Catalog Table Process, for more information.

### Related Topics

- Validating the Sales Catalog
- Running Refresh Denormalized Product Catalog Table Process
- Best Practices for Sales Catalog Setup

### Validating the Service Catalog

After you have published and enabled your catalog, validate that the product groups are displayed in service requests. Use the following procedure.

1. Sign into Engagement Cloud as a service manager or service representative.
2. Navigate to Service Requests and create a service request.
3. In the Create Service Request page, click the Product drop-down list, and click Search.
4. Ensure that your product catalog is displayed in the Select: Product page.

Related Topics

- Running Refresh Denormalized Product Catalog Table Process
- Best Practices for Sales Catalog Setup
6 Setting Up Work Assignment and Routing

Work Assignment: Implementation Concepts

Work Assignment: Overview

In Oracle Cloud, you use the assignment engine to assign resources (for example, service personnel or territory owners) to the business objects they must work on, such as a service request. Being assigned to business objects gives resources and their manager’s visibility into the business object. You also can use rule-based assignment to assign additional resources to objects.

Candidate and Work Objects

When setting up assignment, you must be familiar with two types of assignment objects: candidate objects and work objects:

- Work objects are the business objects that are assigned, for example, service requests.
- Candidate objects are the possible pool of assignment candidates, for example, resources.

Rule-Based Assignment

Rule-based assignment lets you set up additional rules that are used to assign resources to work objects. After you set up the rules containing the conditions that records must meet when resources match the rule conditions, they’re assigned to the object.

For example, you can use rules to assign a certain agent to a certain queue when the customer is located in a specific state or region.

Rule-based assignment requires that you plan your rules, create the rules using the rules UI, and set profile options to configure the assignment behavior, in addition to any scheduled processes that must be run.

Assignment Profile Options

Each of the business objects available in assignment has its own set of profile options that allow you to further configure the application behavior.

Scheduled Processes

Scheduled processes are batch jobs that capture data and allow business objects to act on that data. You must schedule several processes when using assignment.

Assignment Reports

You can use the Diagnostic Dashboard to generate reports about the assigned objects and the volume of territory data involved in assignment.
Assignment Resources
To learn more about assignment in Oracle Cloud, refer to the following resources:

- Related topics: If you’re reading this topic in the Oracle Sales Cloud - Implementing Service in Oracle Engagement Cloud guide, refer to the related topics in the Setting Up Work Assignment chapter.
- Online help: Use the keyword `assignment` to search for the relevant topics.
- Assignment Resource Center: See the Assignment Manager Resource Center page on My Oracle Support (Doc ID 1522958.1) for more resources.

Configuring Assignment: Critical Choices
Assignment is the process of selecting a candidate object and executing the association to a work object. Assignment consists of three phases:

- Setup phase: Setting up assignment processing through assignment configuration.
- Matching phase: Matching rules or mappings are evaluated to find the right assignees from a list of possible candidates.
- Assignment phase: The assignment of matching candidates is handled.

An assignment configuration is predefined for each Sales Cloud application providing assignment processing. This assignment configuration is available from one of the following setup tasks:

- Manage Customer Center Assignment Objects
- Manage Sales Assignment Manager Objects
- Manage Sales Lead Assignment Objects

You can use these setup tasks to add or remove assignment attributes, define the relationship between each work object and candidate object, and define mapping sets and mappings that drive territory-based assignment and rule categories that drive rule-based assignment.

**Note:** Use the Manage Sales Assignment Manager Objects task for opportunity assignment.

The predefined assignment configuration also includes the mapping sets and mappings that drive territory-based assignment.

To best plan the configuration, consider the following points:

- Assignment objects
- Attributes
- Mappings sets and mappings
- Rules

Assignment Objects
An assignment object is a data entity or a collection of data treated as a unit, such as a sales account, an opportunity, or a lead. During assignment configuration, carefully consider which of your business objects require assignment, and create work objects only for those.

A set of assignment objects is predefined for the assignment of territories or resources to accounts, partners, opportunities, leads, and deals.
Attributes

For example, you might want to assign a sales representative (resource) to an opportunity (assignment object), based on the risk level of the opportunity. In this case, you will select the attribute of the opportunity work object that corresponds with risk level, and the attribute of the resource candidate object that corresponds with the name or e-mail address. Selecting these attributes makes them available for mappings and for conditions on your rules. Therefore, ensure that you select the attributes that reflect the criteria that you want to use for matching candidate objects to work objects. Some attributes are predefined as assignment attributes for each assignment object.

Related Candidates

Candidate objects are related to work objects and for each relationship, the appropriate assignment mode (such as matching and scoring) and processing options are predefined. You must not modify these predefined settings except for the No Matches Handling option for the Sales Lead work object. The No Match Handling option controls the assignment behavior when no matching candidate is found. By default, this is set to Remove current assignment. You can change this to Retain current assignment which retains the current candidate assignment when no matching candidate is found. You can also change it to Error which generates an error if no matching candidate is found.

Mappings Sets and Mappings

Assignment mapping sets and their related mappings drive territory-based assignment. The mapping sets determine which mappings are used, and the sequence mapping sets are used in territory-based assignment. The mappings identify the dimensions, attributes, and territory filtering used in the assignment processing. Default mapping sets and their related mappings are predefined.

Rule Categories, Rule Sets, and Rules

The application provides default rule categories. These rule categories identify the type of rule processing being performed, such as matching, scoring, classification or territory. Rule sets group the assignment rules and determine the additional processing performed, such as using scores for each candidate and filtering the candidates assigned to top or random matches. Rules are defined to execute rule-based assignment. Rules are designed to return candidates if they match a set of criteria, are within a defined scoring range, or are of a specific classification.

Create rules using work objects, candidate objects, and attributes that you already established. When designing your rules, carefully consider how you want to match candidates to work objects. For example:

• Would you want resources assigned based on their geographic location, their product knowledge, on the status or score of an object, or a combination of any of these attributes?
• Do you want to match candidates only, or would you like to match candidates and score them?
• In a multiple-candidate scenario, do you want to assign all matching candidates or only those who achieve higher than a specific score?

Consider these questions before creating rules.

Related Topics

• What’s the difference between rule-based and territory-based assignment?

Assignment Mappings
Exporting and Importing Assignment Objects and Rules Setup Data: Explained

This topic explains exporting and importing assignment objects and rules setup data, along with the points to consider while moving the setup data.

Almost all application implementations require moving functional setup data from one instance into another at various points in the life cycle of the applications. For example, in an enterprise application implementation, a development or test instance is first deployed before deploying a production instance. You can move functional setup configurations for assignment objects or assignment rules from one application instance into another by exporting and importing configuration packages from the Manage Configuration Packages page.

To export and import assignment setup data, start by defining an implementation project for the required assignment setup task. The following are some of the examples of assignment tasks:

- Manage Customer Center Assignment Objects
- Manage Service Assignment Manager Objects

The Manage Configuration Packages setup task exports the assignment objects or rules setup data.

A configuration package contains the setup import and export definition. The setup import and export definition is a list of setup tasks and their associated business objects that identifies the setup data for export as well as the data itself. You generate the setup export and import definition by selecting an implementation project and by creating a configuration package. The tasks and their associated business objects in the selected implementation project define the setup export and import definition for the configuration package. In addition, the sequence of the tasks in the implementation project determines the export and import sequence.

You can export a configuration package once you create it, or at any time in the future. During export, appropriate setup data is identified based on the setup export definition, and is added to the configuration package. The setup data in the configuration package is a snapshot of the data in the source application instance at the time of export. Therefore, publish the assignment objects and rules before exporting them. After the export completes, you can download the configuration package as a zipped archive of multiple XML files, move it to the target application instance, and upload and import it. Review and publish the assignment objects and rules setup data in the target application instance to make them available for assignment processing.

See the chapter about importing and exporting setup data in the Using Functional Setup Manager guide for more details.

Exporting and Importing Setup Data: Points to Consider

Based on your implementation, you can follow different approaches while exporting and importing assignment setup data. Consider the following points:

- If your implementation is using only territory-based assignment, the implementation project must include only the Assignment Objects setup tasks.
- If your implementation is using territory-based assignment with rule filtering or rule-based assignment, the implementation project must include both the Assignment Objects and Assignment Rules setup tasks.
- If you are not sure whether your implementation is using territory or rule-based assignment, it is recommended that you include both Assignment Objects and Assignment Rules setup tasks in the implementation project.
- Retain the default sequence for the tasks and business objects.

The application lets you delete assignment objects, assignment attributes, rule categories, rule sets, rules, and conditions in an environment, for example test. If that setup data is exported, and then imported into another environment, for example production, the data in the target database is not removed.
If your implementation plans to import and export setup data for assignment objects and assignment rules, ensure not to delete assignment objects, rule categories, rule sets, and rules. Set them to inactive in case you want to delete them. Additionally, do not delete assignment rule conditions. Instead, set the rule to inactive and then recreate the rule excluding the condition that is no longer needed.

**Related Topics**
- Configuration Packages: Explained

### Mapping Set Components: How They Work Together

Assignment mapping sets and their underlying mappings drive territory-based assignment. This topic explains how these components work together in assignment processing.

The following figure identifies the Sales Cloud work objects that have associated default mapping sets and mappings. Mapping sets and mappings drive territory-based assignment, and the work object attributes map to the territory candidate object dimensions and attributes.

![Mapping Set Components Diagram](image)

### Mappings

The mappings identify the dimensions, attributes, and territory filtering used in the assignment processing. Default mapping sets and their related mappings are predefined for account, lead, partner account, deal, and opportunity revenue assignment. This predefined mapping assumes that opportunities, leads, sales, partners, accounts, and deals use the same territory hierarchy.

Each predefined mapping set has between 9 and 16 mappings that determine the information about the object, such as the account industry or the sales lead product, and how each is mapped to a dimension or attribute on the territory.

You can create additional mappings using the work objects, candidate objects, and attributes that you already established.
Mapping Sets

Mapping sets enable the grouping of mappings so that you can create more than one mapping for each combination of work object and candidate object. The mapping set concept is used only with territory-based assignment and territory-based assignment with rule filtering. Mappings sets are predefined for accounts, leads, opportunities, partner accounts, and deals. When managing assignment objects, the user can define additional mapping sets, each of which contains multiple mappings, for each combination or work object and candidate object.

Mapping Types

There are three types of assignment mapping:

**Dimension Mapping:** Dimension mappings must be used when the work object and candidate object attributes in the comparison are dimension attributes, such as Product. When creating the mapping, use the Function Code field to specify a unique identifier for the dimension. Generally attribute mappings are used when the work object and candidate object attributes in the comparison are non-dimensional attributes. But there are attribute mappings predefined to match the geography and account information about the account with the geography dimension, and account inclusions or exclusions respectively.

When creating the mapping, the Function Service and Function Code are only needed if a translations function is used. The function code field is used to specify a unique identifier for the attribute, and this identifier is passed to the translation function.

An example is assigning territories to opportunity revenue lines based on the product associated with the revenue line. In this case, dimension is selected as the mapping type. The candidate object low attribute and high attribute correspond to the names of the low sequence and high sequence attributes for product on the territory. The work object low attribute and high attribute correspond to the names of the low sequence and high sequence attributes for product on the revenue line.

**Attribute Mapping:** This mapping enables you to compare and match attribute values between a work object attribute and a candidate object attribute. When the value of the candidate object attribute matches the work object attribute, the candidate is selected. Attribute mappings are typically used when the work object and candidate object attributes in the comparison are non-dimensional attributes. This type of mapping is also used to capture the mapping between hierarchical dimensions account and geography.

For example, consider a lead work object with a Partner Identifier attribute and the territory object with Partner ID attribute. The selection criterion is: select Sales Lead Territories where Sales Lead Territory.Partner Identifier equals Sales Lead.Lead Partner Identifier. The assignment engine will use this mapping data to construct a query on the candidate object that is equivalent to the selection criteria.

**Literal Mapping:** Literal Mapping is used almost exclusively to filter the candidate objects. This form of mapping enables the comparison of candidate attributes against a specific value chosen by the user. The assignment engine will compare the mapped candidate object attribute against the specified literal value. For example, select the Territory Candidate object that has the attribute Coverage Model that equals the value PARTNER_CENTRIC.

> **Note:** For Literal Mappings, ensure that the value entered corresponds to the Lookup Type Value code, not the meaning.

Assignment Processing Using Mapping Sets and Mappings

When designing your mappings, carefully consider the dimensions and attributes you use in your territory structure and how you want to match these territory candidates to work objects. Also consider the shape of the information used in the territory structure; this may affect the sequence of each mapping. A sequence can be entered for each mapping set which is used to determine the order in which these mapping sets will be used in the territory-based assignment processing. The sequence of the dimension mappings used in territory matching can affect performance. The most selective mapping should be given the lowest sequence number. By default, this dimension is the Geography Dimension. By using the lowest sequence number, it is
performed earliest in the matching process, which results in the smallest number of territory matches. Mappings that do not have a sequence are used together at the end of the matching process.

Sometimes the mapping set sequence does not matter. For example, there are two predefined opportunity revenue assignment mapping sets. When the first mapping set is used, it finds matching territories based on the information about the opportunity/opportunity account, and the territory information. Then the second mapping set is used which matches territories based on the opportunity/opportunity partner information and the territory information. The order of the mapping sets are interchangeable; regardless of which mapping set is used first, the resulting territories that match will be the same.

In the case of leads, the mapping set sequence is important as the territories matched using the first mapping set may result in a primary partner being added to the lead. This information is significant to the territory matching performed using the second mapping set.

Mapping sets can be made conditional to control whether the mapping set is used or not used during assignment processing. For example, the partner channel manager territory assignment mapping set conditional attribute is set to the value RevenuePartnerId. During the assignment processing of a revenue line, if the Revenue PartnerId attribute for that revenue line contains a value, then this mapping set will be used in territory matching processing.

An indicator in the Related Candidates region controls whether to merge the matching assignment candidates identified from processing each set of mappings. This indicator is used to drive the merging of matching candidates when multiple mapping sets are used in assignment processing. If the box is checked, then the candidates are merged. The default is not to merge the candidates.

In most implementations, the predefined mapping sets are sufficient. But mapping sets can offer some flexibility if user-defined assignment processing is needed.

Assignment Rules

Assignment Mappings: Examples

For territory-based assignment, you must create work object to candidate object mappings while creating the assignment object. These mappings are used to make candidate assignments. The scenarios in this topic illustrate creating the following mapping types:

- Attribute mapping
- Dimension mapping
- Literal mapping

Creating an Attribute Mapping

Assign territories to an opportunity revenue line when the territory line of business is the same as the opportunity line of business. To create an attribute mapping:

1. Create the following mapping:
   - Work object - Revenue
   - Candidate object - Territory
2. Select the territory when the attribute territory line of business code is equal to the revenue line of business.
3. Enter a value for the sequence which determines the order in which the application uses the mapping when matching territories.
Creating a Literal Mapping

Literal mappings are a way of filtering the matched territories based on specific values of a territory attribute. Find only territories that have an account-centric coverage model assigned to each revenue line. For example, territory coverage model equals SALES_ACCOUNT_CENTRIC.

Note: Literal mappings use the code value for lookup-based fields, and not the meaning value.

To create this literal mapping:

1. Select the mapping type **Literal** and optionally enter a sequence value, which determines the order in which the application uses the mapping when matching territories.
2. Select the candidate object **Territory**.
3. Select the candidate object attribute that will be used for filtering. For example, Coverage Model.
4. Select the operator value **Equals**.
5. Select the literal value. In this example, only sales account centric territories must be assigned to revenue lines, so the Literal Value entered corresponds to the code value for the coverage model. For example, SALES_ACCOUNT_CENTRIC.

Defining Service Assignment Rules: Explained

Assignment rules are used to automatically assign service requests to queues when the service requests are created or updated. Rules can be run on a schedule to automatically assign service requests to a queue when a service request is created or updated.

Service requests are treated as work objects and Queues are treated as candidate objects. Define your rules to select the best candidate (queue) for each work object (service request).

Note: Service request assignment rules are defined using Rule-Based Assignment. Territory-Based Assignment does not apply to service requests.

Defining service request assignment rules requires some forethought. Consider the following before you define service request assignment rules:

- The attributes of queues you want to use as criteria for your rule assignments.
- The attributes of service requests you want to use as criteria for your rule assignments.
- The rule sets you want to create.
- The rules to include in each rule set.

Note: You must define queue candidate objects before you start defining rules for service request assignment objects.
Manage Service Assignment Objects
In this procedure, you select attributes from the service request assignment object that you want to make available in your rules. This procedure however, is not mandatory, as ready-to-use fields are provided for all of the objects.

To manage service request assignment objects:

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Communication Channels functional area.
5. Select the Manage Service Assignment Objects task.

The Manage Service Assignment Objects page appears.

6. Add the queue attributes you want to be available when setting up your rules:
   a. Click Queue.
   b. Click the Attributes tab.
   c. Add attributes to the list by clicking the plus icon, then selecting the View Object Attribute from the dropdown list.
   d. Click Save.

7. Add the service request attributes you want to be available when setting up your rules:
   a. Click Service Request.
   b. Click the Attributes tab.
   c. Add attributes to the list by clicking the plus icon, then selecting the View Object Attribute from the dropdown list.
   d. Click Save.

Manage Service Assignment Rules
In this procedure, you define the rules for service assignment.

When assigning work items to Queues, follow these guidelines when defining your matching rules:

- The rule set must be defined with Number of Candidates = 1. The application enables only one queue to be assigned to a service request.
- You have the option to select or deselect the Use Score option on a rule set. If you select Use Score, then for every rule in the rule set, you must indicate the amount to increase the score when the rule is true. You must then associate the rule set to queues that receive that score. All of the rules in a rule set are executed, and the Queue with the highest total score is selected.
- If the rule set has multiple rules and you did not select the Use Score option, you must define the criteria for each rule to be mutually exclusive from other rules in the rule set. This ensures that the resulting Queue assigned by the application is predictable in all situations.

Operators are used to define the conditions for service request assignment rules. The following table lists the operators.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Value</th>
<th>Used in hierarchy?</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>The value of a field equals a specific value.</td>
<td>Single</td>
<td>No</td>
<td>If the rule condition is set as Service Request Severity Equals High, only a service request with the severity value</td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td>Value</td>
<td>Used in hierarchy?</td>
<td>Example</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Does not equal</td>
<td>The value of a field does not equal a specific value.</td>
<td>Single</td>
<td>No</td>
<td>If the rule condition is set as Service Request Severity Does not equal High, a service request with severity value of Low or Medium matches the condition.</td>
</tr>
<tr>
<td>In</td>
<td>The value of a field is one of a list of values.</td>
<td>Single or Multiples</td>
<td>No</td>
<td>If the rule condition is set as Service Request Problem type Code In Docs, Product, a service request with the problem type code value of Docs or Product matches the condition.</td>
</tr>
<tr>
<td>In including children</td>
<td>The value of a field is one of a list of values.</td>
<td>Single or Multiples</td>
<td>Yes</td>
<td>If the rule condition is set as Service Request Product Group In including children Apple, any service request with the Apple product group or any of its children, such as iPhone, iPhone 6s, iPhone 7, iPad, iPad1, and iPad 2 match the condition.</td>
</tr>
<tr>
<td>Not in including children</td>
<td>The value of a field is not in a list of values. This value is only relevant for category or product fields which are hierarchical. Indicates that the rule applies if the specified attribute value matches the top level of the attribute. This option does not include the attribute values of the children of the current attribute.</td>
<td>Single or Multiples</td>
<td>Yes</td>
<td>If the rule condition is set as Service Request Product Group Not in including children iPad, a service request with the Product Group value of iPad or any of its children, such as iPad1 or iPad2 do not match the condition. However, a Service request with the Product Group value of Apple, iPhone, or iPhone 6s matches the condition.</td>
</tr>
<tr>
<td>Not in</td>
<td>The value of a field is not in a list of values.</td>
<td>Single or Multiples</td>
<td>No</td>
<td>If the rule condition is set as Service Request Problem Type Code Not in Docs, Product, a service request with the problem type code value of User does not match the condition. However, a service request with a problem type code...</td>
</tr>
<tr>
<td>Operator</td>
<td>Description</td>
<td>Value</td>
<td>Used in hierarchy?</td>
<td>Example</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------</td>
<td>-------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Is blank</td>
<td>The value of a field does not contain a value.</td>
<td>N/A</td>
<td>No</td>
<td>If the rule condition is set as Service Request Internal Priority Code Is blank, a service request without an internal priority code value is a match.</td>
</tr>
<tr>
<td>Is not blank</td>
<td>The value of a field contains any value.</td>
<td>N/A</td>
<td>No</td>
<td>If the rule condition is set as Service Request Account Is not blank, a service request with an associated account is match, but not a service request without an account.</td>
</tr>
</tbody>
</table>

To manage service request assignment rules:

1. In the **Communication Channels** functional area, select the **Manage Service Assignment Rules** task.

   The **Manage Service Assignment Rules** page appears.

2. Select a **Category** based on the following.

   a. **Generic Queuing Rules**: Rules set for all types work items.
   b. **HR Service Request Queuing Rules**: Rules set for service request for HR help desk
   c. **Service Request Queuing Rules**: Rules set for service request for CRM

3. Create a new rule set by clicking the plus icon in the **Rule Sets** work area, and then enter the required information.

4. Create rules for the rule set by clicking the plus icon in the **Rules** work area.

   The **Create Rule** screen appears.

5. Enter a name for the rule in the **Name** field.

6. From the **Rule Applies If** drop-down list, select **Any conditions met**.

7. Add a condition by clicking the plus icon in the **Conditions** work area, and then define the required attribute.

   If an attribute is hierarchical, such as Category Name and Product Group, **Not In Including Children** and **In Including Children** operators are displayed as choices. For more information about the operators, see the table with list of operators.

8. (Optional) Add additional conditions.

9. In the **Action: Assign Queue** section, click the plus icon to select a queue.

10. In the **Select and Add: Queue** window, search for and select a queue.

11. Click **OK**.

12. Click **Save and Publish** to publish the assignment rules.

The service assignment has been defined.

*Note:* Republish the assignment rules each time the rule is changed. You also need to republish the rules each time the associated queue is deleted, enabled, or disabled.
Assignment Rule Components: How They Work Together

The rule category, rule sets, and rules are components that work together to determine how the assignment engine processes rule-based assignments for work objects.

Rule categories are predefined for each object leveraging assignment rules. Each predefined rule category determines the type of rule processing performed, for example, matching, scoring, and classification.

Depending on the rule category selected, rule sets may allow filters to be used to determine whether all matches are assigned, or a random number of matches. Additionally, a score may be used to allow further filtering of the matching candidates, such as the top X candidates or all above or equal to a minimum score.

At the rule level within a rule set, the action determines the behavior when a rule is evaluated as true. The rule action option works in conjunction with the rule category selected.

The following table describes how the rule set components work together.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Use Score</th>
<th>Filters</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching or Territory</td>
<td>X</td>
<td>All</td>
<td>Increase Score By x for each matching or selected candidate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All Above Minimum Score (set Minimum Score value)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Random (set Number of Candidates value)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Top X (set Number of Candidates value)</td>
<td></td>
</tr>
<tr>
<td>Matching or Territory</td>
<td>Not applicable</td>
<td>All</td>
<td>Assigns the matching or selected candidates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Random (set Number of Candidates value)</td>
<td></td>
</tr>
<tr>
<td>Scoring</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Increase Score By x</td>
</tr>
<tr>
<td>Classification</td>
<td>Not applicable</td>
<td>Not applicable</td>
<td>Set Value To x</td>
</tr>
</tbody>
</table>

Rule Category and Rule Set

The rule category selected for the rule set determines the type of rule-based assignment processing to be performed. For example, if you select the rule category named Sales Lead Resource Rule Category, the candidates that match the conditions of the rules evaluated as true by the assignment manager are assigned to the work object. The number of matching candidates that are assigned to the work object is determined by the rule set filter settings. Only one rule category can be associated with each rule set.

A rule category is predefined for each type of rule-based assignment processing supported by each Sales Cloud object. For example, the rule category named Sales Team Member Recommendation Default Rule Category is predefined for resource rule-based assignment of Opportunities, and Sales Lead Resource Rule Category is predefined for resource
rule-based assignment of Leads. Similar rule categories can be predefined for territory rule filtering for revenue lines, territory rule filtering for leads, lead scoring, lead raking, and lead qualification.

Rule categories are created and edited through the Manage Assignment Objects setup task for the relevant application. A rule set contains rules that belong to a specific rule category.

**Use Score**

The **Use Score** option determines whether a score is used when identifying matching candidates. The number of matching candidates that are assigned to the work object is also determined by the rule set filter settings.

**Filter Settings**

The filter settings are used in conjunction with some rule categories and the rule set **Use Score** option. The filters allow you to indicate how many matching candidates you want to assign to the work object. When set to **All Above Minimum Score**, all of the matching candidates above a particular score are assigned to the work object. Set the score in the **Minimum Score** field.

When set to **Top X**, a number of matching candidates with the highest scores are assigned to the work object. Use the **Number of Candidates** field to specify how many top matching candidates to assign.

When the filter is set to **Random**, a random selection of matching candidates is assigned to the work object. When the rule set **Use Score** option is selected, and the filter is set to **Random**, a random selection of matching candidates with the highest scores is assigned to the work object. Use the **Number of Candidates** field to specify how many random matching candidates to assign.

**Rules**

One or more rules may be defined for each rule set. Each rule is the distinct set of criteria that is evaluated and candidates or scores that are eligible to assign if the conditions are met. The rule action may apply if all conditions are met, or any conditions are met.

The assignment rule administration allows more than one user at a time to create or update rules that belong to the same or different rule sets or categories. For example, if User A is currently updating assignment rules for the **Sales Lead Resource Rule Category** rule category, then User B can update assignment rules for that same rule category or another rule category at the same time.

**Action**

The action set at the rule level determines the action that is performed when a rule is evaluated as true.

If defining rules to assign resources to an object, you can search for and select the specific resources to be assigned when the rule conditions are evaluated as true.

When a matching rule category is selected, for example, Sales Lead Resource Rule Category, the rule action assigns the matching candidates. If a rule with that action is evaluated as true, the candidates that match the conditions for that rule are assigned. The filter setting at the rule set level determines whether all matching candidates are assigned (All), or a random number of matching candidates are assigned (Random).

When a matching rule category and the **Use Score** option are selected, the rule action increases the candidate score by the specified value. If a rule with that action is evaluated as true, the candidates that match the conditions for that rule get the value in the **Action** added to their score. For example, you select Sales Team Member Recommendation Default rule category and the **Action** for one of the rules in that set is **Increase Score By 10**. If that rule is evaluated as true, the resources that match the conditions for that rule get 10 added to their scores. The scores are cumulative, so if any of the resources that matched the conditions in the rule in the example also match the conditions for other true rules in the set, those resources get additional values added to their current score of 10. The filter setting at the rule set level determines
whether all matching candidates are assigned (All), or all matching candidates above a specified score are assigned (All Above Minimum Score), or a random selection of matching candidates with the highest scores are assigned (Random), or a number of matching candidates with the highest scores are assigned (Top X).

When a classification rule category is selected, the rule action is Set Value To Value Name. For example, the rule category is Sales Lead Rule Qualification Rule Category, the action for one of the rules in that set is Set Value to Qualified. If that rule is evaluated as true, the Status for the lead being classified is set to Qualified.

When the rule category Sales Lead Scoring Rule Category is selected, the rule action is Increase Score By Score Value. If a rule with that action is evaluated as true, the value in the action is added to the score of the work object associated with the rule set. For example, if the action for one of the rules in that set is Increase Score By 20, and that rule is evaluated as true, the score for the Lead is increased by 20.

Note: When you are creating or updating assignment rules, you must click Save and Publish to ensure that your changes are live and included in the assignment processing.

Creating Assignment Rules: Examples

Assignment rules are created using rule sets, rules, conditions, and actions. The assignment engine uses your rules to evaluate and recommend candidate assignments for specified work objects. This topic provides scenarios to illustrate the different types of rules you can create.

Creating Lead Qualification Rules

In this scenario, you want to create rules to classify leads as qualified if the following attributes are set as specified:

- Lead Customer is sales account.
- Lead Product is set to Is Not Blank.
- Lead Score is greater than 150.

To create a rule to classify leads as qualified:

1. Sign in as a setup user or sales administrator.
2. Click Navigator Setup and Maintenance.
3. On the Setup page select the Sales offering.
4. Select the Leads functional area.
5. In the Show field select All Tasks.
6. Search for and navigate to the assignment configuration setup task for the relevant object:
   - For sales lead and deal assignment, go to the Manage Sales Lead Assignment Rules task. This is the task used as an example in this topic.
   - For the Opportunity functional area assignment, select the Manage Opportunity Assignment Manager Rules task.
7. In the setup task page, select the category for the appropriate assignment flow, in this case Sales Lead Qualification Rule Category.
8. Click the Add Row icon to create a rule set for the predefined rule category Sales Lead Qualification Rule Category.
9. Create a rule with the three conditions that match the attribute settings for a lead to be considered a qualified lead:
   - Lead Product: Select the lead attribute Primary Product ID. Select the Is Not Blank operator.
   - Lead Customer: Select the lead attribute Sales Account Indicator, and then select the Equals operator. Enter the value of Y.
Lead Score: Select the lead attribute **Score**, and then select the **Greater Than** operator. Enter the value of 150.

10. In the Actions region, select **Qualified** from the Return the Candidate Value As Qualified list.

Creating Lead Scoring Rules

In this scenario, you want to create a scoring rule to:

- Increase lead scores by 150 if the lead attribute Lead Time Frame is set to 3 months.
- Increase lead scores by 100 if the following attributes for leads are set as specified:
  - Budget Status is Approved
  - Budget Amount is greater than 500000

To create this scoring rule:

1. Create a rule set for the predefined rule category Sales Lead Scoring Rule Category.
2. Create the first rule with the conditions that match the attribute settings you want a lead to have in order to add 150 to its score:
   - Choose the object **Sales Lead** and attribute **Time Frame**, and then select the **Equals** operator. Select 3 months.
   - Enter the action as **Increase the Score by 150**.
3. Similarly, create your remaining rule for the budget attributes and action to **Increase the Score by 100**.
   a. Add the first condition: Choose the object **Sales Lead** and attribute **Budget Status**, and then select the **Equals** operator. Select Approved.
   b. Add the second condition: Choose the object **Sales Lead** and attribute **Budget Amount**, and then select the **Greater Than** operator. Enter 500000.
   c. Enter the action as **Increase the Score by 100**.

Creating Matching Candidate with Scoring Rules

Assign different country specialists to opportunities in some European countries based on the country and the risk level of the Opportunity. To create matching candidate with scoring rules:

1. Create a rule set for the predefined category Sales Lead Resource Rule Category and select the **Use Score** option, the filter type of **All Above Minimum Score**, and the minimum score set to 20.
2. Create three rules each with conditions:
   a. Create the first rule with the following condition and actions:
      - Select the object **Opportunity**, and then choose the attribute **Customer Country**. Select the **Equals** operator, and then select **DE**.
      - In the rule action, set **Increase Score By** to 20.
      - Select and add the appropriate resource.
   b. Create the second rule with the following conditions:
      - Select the object as **Opportunity**, and then choose the attribute name **Customer Country**. Select the **In operator**, and then select **FR** and **UK** as condition values.
      - In the rule action, set **Increase Score By** to 20.
      - Select and add the appropriate resource.
c. Create the third rule with the following conditions:
   - Select the object **Opportunity**, choose the attribute name **Risk Level**. Select the **Equals** operator, and then select the value **High**.
   - In the rule action, set **Increase Score By** to **20**.
   - Select and add the appropriate resource.

**Related Topics**
- What’s the difference between rule-based and territory-based assignment?
- Creating Rules to Assign Matching Candidates

**Categories**

**Managing Service Request Categories: Explained**

Service request categories can help identify the nature of issues reported in service requests. For example, categories can help group service requests related to hardware in one category, and service requests related to software in another category. Further categories and child categories can then be created to narrow the type of service request within one of the ordered groupings.

Administrators can create categories and category hierarchies to group and organize service requests depending on their organizational needs. Before creating categories, consider the following:

- Create a list of your top-level categories.
- For each top-level category, create a list of child categories.

To create service request categories do the following:

1. Sign in to the Engagement Cloud application as an administrator or a setup user.
2. Navigate to **Setup and Maintenance**.
3. Click the **Setup** drop-down list and select **Service**.
4. In the **Service Request** functional area, select to display all tasks and select **Manage Service Request Categories** task.
   - The Manage Service Request Categories page appears.
5. Create a top-level category:
   a. Click the **Create Category** drop-down list, and then select **Create Top-Level Category**.
   b. Enter a name in the **Category Name** field.
   c. Specify if the category must be active, by selecting a value in the **Active** drop-down list.
   d. Enter a unique **Short Code** for the category.
   e. Create additional top-level categories, as needed.
6. Create child categories:
   a. From the **Service Request Categories** list, select the top-level category for which you want to create child categories.
   b. Click the **Create Category** drop-down list, and then select **Create Child Category**.
   c. Enter a name in the **Category Name** field.
d. Specify if the category must be active, by selecting a value from the **Active** drop-down list.
e. Create additional child categories, as needed.

The child categories appear indented under the top-level category.

### Managing Work Assignment

#### Assignment of Work: Explained

Work assignment refers to interactions or work items, such as Service Requests being assigned to queues for processing. Work items are assigned to a queue and an agent is assigned to one or more queues. All work items in a queue are handled by the assigned agents.

A work item can be assigned to an agent manually or automatically, based on if the associated queue is automatic or manual. For more information about the types of queues, see Managing Queues: Procedure. Work assignment in an automatic queue is done based on the following aspects.

- Agent capacity
- Availability of the agent
- Presence of the agent
- Severity of the work item
- Age of the work item

**Note:** Omnichannel must be enabled to use the settings for Presence of the agent, Availability of the agent, and Agent capacity.

#### Agent Capacity

Capacity indicates the maximum number of open work items of a certain type that an agent can handle. Capacity is configured at a global level for various types of work items using the **Setup and Maintenance** page. Capacity is calculated separately for real-time work items, such as chat requests, and non-real-time work items, such as SRs. For example, if the global capacity for open SRs is 30, an agent who is working on 30 open SRs is considered to be 100 percent used. If a queue has multiple agents, work item is assigned to the agent with the maximum free capacity of the relevant type. If more than one agent has the same capacity, the work item is assigned to the agent with the maximum overall free capacity. For more information about setting global capacity, see Setting Channel Capacity: Procedure.

#### Agent Presence

Presence indicates if an agent who has signed in to the application is interacting with the application. An agent sets their presence through **Settings and Action** menu. An agent can be online, but may not be available to handle work items. If an agent is busy, no work items are assigned to them. If an agent is not signed in, the presence is set to offline.

#### Agent Availability

Agent availability indicates if an agent is available to handle an interaction or a work item. Once an agent signs in to the application, they can set their availability to handle an interaction. An agent can explicitly specify their availability for real-time and non-real-time work items. This means that an agent can specify if they are available for chat or for SR or for both using the icons on the Omnichannel toolbar.
If an agent presence is set to Busy, but availability is set to non-real-time work items, an agent can still be assigned SRs. However, if an agent is available to handle real-time work items, presence is automatically set to online.

Service Request Severity and Age

Severity indicates the priority of a service request. An SR is assigned to an agent based on its severity. This means that an SR with a higher severity is taken up earlier for resolution. An administrator can set the values for severity in the Setup and Maintenance page. For more information, see Adding Service Request Severity Values: Procedure.

Age of an SR is also a factor while assigning work. An SR that is waiting in the queue for a longer period of time is given higher priority.

Related Topics
- Managing Queues: Procedure

Set Rules for Queue Assignment: Procedure

Assignment rules are created using rule sets, rules, conditions, and actions. The assignment engine uses your rules to evaluate and recommend candidate assignments for specified work objects. For example, you can assign all service requests with a certain product code to one queue, or you can assign a service request (SR) to a critical queue if the severity of the SR is High. You can also create assignment rules to assign real-time work items, such as chat notifications to queues. A real-time and a non-real-time work item can be assigned to the same queue.

Note: You can assign a work items only to a single queue. Hence, you can’t define rules that each assign the same work items to different queues.

To create a service assignment rule, you perform the following tasks.

1. Sign in to Oracle Cloud Applications as a service administrator.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Communication Channel functional area, and then select Manage Service Assignment Rules task.
5. In the Manage Service Assignment Rules page, select Service Request Queuing Rules option for service requests or Generic Queuing Rules for chat, from the Category drop-down list.
6. Create or select a rule set.
7. Create a rule.
8. Specify an assignment condition. This condition is evaluated before a rule based assignment is made.
9. Select a queue to which an SR is assigned if the condition is met.

Creating a Rule Set

One rule set can contain multiple rules. To create a rule set, click plus and specify the required values.

Creating a Rule

One or more rules can be created within a rule set. Click plus in the &lt;Rule set&gt;: Rules section. To create a rule, do the following.

1. In the Create Rule page, specify rule name.
2. (Optional) Specify a description, effective start and end dates of the rule. If you do not select an effective start and end date, the rule will come into immediate effect and last indefinitely.
3. (Optional) Select the Inactive option if you want to enable the rule at a later date.
Specifying a Condition and an Action

After specifying the rule details, specify one or more conditions and select a queue to which the SR is assigned if the condition is satisfied. When the specified condition is satisfied, the action of assigning the SR to a queue is performed. For example, if the severity of the service request is equal to high, assign the SR to the Critical_Queue. To add a condition, do the following.

1. Select an option from the Rule Applies If drop-down list. You can choose to perform the action if all of the conditions are met or if one of the conditions is met.
2. Click plus to add a condition.
3. Select the object, an attribute, an operator for the condition, and a value. For example, Service Request is the object, Severity is the attribute of the object, Equals is the condition, and High is the value. Hence, the condition is If Service Request Severity is equal to High. To create a rule for chat, select Any from the Object drop-down list.
4. Next, add an action to be performed when the condition is satisfied. Click plus to select and add a queue.
5. Search for a queue, select the required queue from the list, and click Done to add the queue to the action.
6. Click Save and Close.

Adding Service Request Severity Values: Procedure

Service request severity indicates the priority of a service request. An SR of a higher severity bypasses other SRs of lower priority in a queue, and is taken up for resolution sooner.

An administrator can set the severity values in Oracle Cloud application by using the Setup and Maintenance page. To add severity values, perform the following procedure.

1. Log on to Oracle Cloud applications as an administrator.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Service Request functional area, and then select the Manage Service Request Severities task.
5. In the Manage Service Request Severities page, ORA_SVC_SR_SEVERITY_CD lookup type contains the lookup codes for various severities. To add a new severity value, click the plus icon and specify the following values.
   - **Lookup Code**: Enter the new lookup code. For example, High Priority.
   - **Severity Ranking**: Enter a value that depicts the priority level. An SR Severity with a lower Severity Ranking value will be considered as having higher priority when SRs are routed to agents. For example, a severity ranking of 2 is considered higher in priority when compared to a severity ranking of 5.
   - **Enabled**: Select to enable the severity value. Disabled severities are not displayed on the SR edit page.
   - **Start Date** and **End Date**: Specify a date range within which the severity value is valid. If you do not specify a date, the severity value remains valid forever.
   - **Meaning**: Enter an optional meaning for the severity value. For example, for Priority 1, specify the Meaning as Critical.
   - **Description**: Enter an optional description.
   - **Tag**: Specify a tag that is used to differentiate an SR of a certain severity from the others. For example, specify Tag for Priority 1 as COLOR=#ff0000,#FFFFFF to change the color of a Priority 1 SR In this case, a Priority 1 SR is displayed in red text with white background.

Note: The Lookup Code is a text value and must be used for display purpose only. The value that you specify in the Lookup Code field does not affect the actual severity of an SR, which is determined by the Severity Ranking field.
6. Add more lookup codes and when done, click **Save and Close**.

Enabling Omnichannel

Enabling Omnichannel: Procedure

Automatic routing of work to agents is done through Omnichannel. Hence, if you want the work assignments to agents to be automatically routed, you must enable Omnichannel.

> **Note:** To view the Omnichannel tasks, you must enable the **Omnichannel Routing** feature under the **Communication Channel** offering. For more information about enabling offerings and features, see "Configuring Offerings: Procedure" in the Using Functional Setup Manager guide.

To enable Omnichannel and its notifications, you must enable the profile options specified in the following table.

<table>
<thead>
<tr>
<th>Profile Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_ENABLE_OMNI</td>
<td>Enable Omnichannel.</td>
</tr>
<tr>
<td>SVC_OMNICHANNEL_BROWSER_NOTIFICATION_ENA</td>
<td>Enable browser notifications about work assignments.</td>
</tr>
<tr>
<td>SVC_OMNICHANNEL_DESKTOP_NOTIFICATION_ENA</td>
<td>Enable desktop notifications about work assignments.</td>
</tr>
<tr>
<td>SVC_OMNICHANNEL_BROWSER_NOTIFICATION_AUTO_INT</td>
<td>Enable the automatic dismissal of browser notifications.</td>
</tr>
<tr>
<td>SVC_INTERACTION_RETENTION_DAYS</td>
<td>Set the number of days to keep Interactions before they’re purged from the database. Must be set to 1 or greater in order for the job to run.</td>
</tr>
<tr>
<td>SVC_EVENTS_RETENTION_DAYS</td>
<td>Set the number of days to keep Events before they’re purged from the database. Must be set to 1 or greater in order for the job to run.</td>
</tr>
<tr>
<td>SVC_INTERACTION_DISPLAY_DAYS</td>
<td>Set the interactions to be viewed from the specified number of days from the past.</td>
</tr>
</tbody>
</table>

To enable Omnichannel, perform the following steps.

1. Sign in to Oracle Cloud applications as an administrator.
2. In the **Setup and Maintenance** work area, select the **Service** offering.
3. Click **Setup** in the **Administration** section.
4. In the **Setup: Service** work area, select the **Communication Channels** functional area, and then select the **Manage Omnichannel Profile Options** task.
5. In the **Manage Omnichannel Profile Options** window, select the **SVC_ENABLE_OMNI** profile option.
6. In the **SVC_ENABLE_OMNI: Profile Values** section, set the **Profile Value** for the **Site Profile Level** as Yes.
7. Click **Save**.
To enable notifications, repeat steps 5 to 7 for the following profile options:
- SVC_OMNICHANNEL_BROWSER_NOTIFICATION_ENA
- SVC_OMNICHANNEL_DESKTOP_NOTIFICATION_ENA

Omnichannel is now enabled and all the work assignments are automatically routed to agents.

Related Topics
- Configuring Offerings: Explained
- Configuring Offerings: Procedure

Setting Channel Capacity: Procedure

Channel capacity indicates the maximum number of active interactions in a channel, that can be handled by an agent. When the number of active interactions in an agent’s queue reaches the specified capacity, the agent is considered to be 100 percent occupied. Agent capacity for a non-real-time channel such as service requests (SRs) is independent of the capacity for real-time channels, such as chat. For example, consider that an SR capacity is set to 30 and chat capacity is set to 2. An agent who is handling 15 open SRs and 2 chats is considered to be 50 percent occupied for non-real-time work and 100 percent occupied for real-time work.

Active non-real-time interactions are determined by the qualifying statuses. Any interaction that is in one of the specified qualifying statuses is considered to be an active interaction.

To set channel capacity, perform the following steps.

1. Log on as an administrator to Oracle Cloud applications.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Communication Channels functional area, and then select the Manage Capacities task.
5. The Manage Capacities page displays capacities for real-time and non-real-time channels. To change the total channel capacity, modify the default values in the Capacity fields for the channels.
6. To change the capacity of a non-real-time work do the following.
   a. In the Work Assignments section, modify the default value in the Capacity field to a new value.
   b. The status in the Qualifying Status Values column indicates the status of the work items that determine the capacity. For example, if the status of set to New, In Progress, only the work items that are in the specified status add up to the total capacity. To specify the active statuses, click the status value.
   c. In the Qualifying Status Values window, select a status from the No Effect on Workload list and move it to the Adds to Workload list to qualify the status as active.
   d. Click Apply.
7. To change the capacity of a real-time work, change the value in the Capacity field in the Communication Channels section.
8. In the Qualifying Status Value window, add a new status to Adds to Workload list and click Apply.
9. Click Save and Close.

Configuring Presence and Availability Privilege: Explained

To set the presence and availability, the agent must be given the Manage Omnichannel Presence and Availability privilege. This privilege, by default is available to the Customer Service Representative, Customer Service Manager, and
Administrator job roles. Hence, to grant the Manage Omnichannel privilege to any other role, you must do one of the following in the Security Console.

- Create a role and provide the privilege to the role.
- Copy one of the default job roles with the privilege and create a role to add the privilege.

After you add the privilege to the new role, associate a user to the new role. For more information about copying job roles, see Oracle Sales Cloud Securing Oracle Sales Cloud guide.

Related Topics
- Copying and Editing Duty Roles: Procedure

FAQs About Service Requests

What happens if a newly created service request isn't assigned to a queue?
If a service request is created without an associated queue, and Omnichannel is enabled, then the routing feature selects a queue. If the queue is automatic, the routing feature assigns the associated agent for the service request. If an SR is created with an associated queue, the routing feature selects only the agent.

What happens if an associated agent is removed from an open service request?
If an associated agent is removed from an open service request that’s in status New or any other active SR statuses that are specified in the Manage Capacities page, the routing feature selects a new agent for the service request from the resource pool of that queue, based on the capacity of the associated agent. For more information about capacity, see Setting Channel Capacity: Procedure. A new agent is selected only if Omnichannel is enabled and the queue is automatic.

What happens if an open service request is disassociated from the current queue?
If an open service request is disassociated from an existing queue, the routing feature assigns the SR to a new queue, provided Omnichannel is enabled. The SR is assigned to an associated agent if the queue is automatic and any assignment rules are rerun. If the disassociated SR is in status New or any other active status specified in the Manager Capacity page, agent capacity is calculated before assigning the SR to an agent. If you disassociate a service request, so that rerouting to takes place, the Assigned To value must be cleared.

What happens if a closed service request is reopened?
If the service request (SR) that is reopened is associated with an agent already, the SR is assigned to that agent. If it isn't assigned to any agent, and Omnichannel is enabled, the SR is assigned to a queue based on the associated assignment
rules. If the queue is automatic, then the SR is assigned to an associated agent. Any SR that isn’t in New or In Progress status is considered to be in Closed status. SRs are assigned based on the set channel capacity. For more information about capacity, see Setting Channel Capacity: Procedure.

How can I set the Assign Service Request to Queue on create option?

Sign in to the application and open the Setup and Maintenance page. Select the Service offering, and click Setup in the Administration area. Select Service Request in the Functional Areas column, and select Manage Service Request Profile Options in the Task list. Select SVC_ASSIGN_TO_QUEUE_ON_CREATE, and select Yes in the Profile Value drop-down list. Click Save and Close.

Alternatively, sign in to the application and open the Setup and Maintenance page. Enter Manage Service Request Profile Options in the search box, and click the Search icon. Select Manage Service Request Profile Options from the search list. Select SVC_ASSIGN_TO_QUEUE_ON_CREATE, and select Yes in the Profile Value drop-down list. Click Save and Close.

Note that this profile option is ignored when Omnichannel is enabled. If Omnichannel is enabled, even if the profile option SVC_ASSIGN_TO_QUEUE_ON_CREATE is set to False then the SR is assigned to a queue when creating the SR.

What happens if I mark an assignment object or one of its attributes as inactive?

When you mark an assignment object as inactive, the selected work or candidate assignment object isn’t available for assignment processing. When you mark an assignment attribute as inactive, the selected work or candidate object attribute isn’t available for assignment processing.

Note: The object or attribute can’t be set to inactive if there is a mapping set, mapping, or rule defined using the object or attribute.
7 Setting Up Communication Channels

Inbound Message Processing: Explained

An inbound message is created when you receive an incoming service email from a customer or a partner. The following figure illustrates the process flow for an incoming customer email:

[Diagram of the process flow for handling inbound messages]
The following steps describe the process flow for an incoming customer email:

1. A customer or a partner sends an email.
2. An inbound message filter is applied to the incoming message.
3. If the message is accepted, an inbound message is created, and any associated attachments are extracted.
4. The message is verified to see if the message is related to an existing Service Request (SR).
   - If the inbound message is related to an existing SR:
     - The related SR is identified. If the related SR can be edited, the email content and attachments are added to the SR. If the original SR can't be edited, a new SR is created.
   - If the inbound message is not related to an existing SR, a new SR is created.
5. The email ID of the sender is validated against the customer or partner records in the database.
   - If the email ID of the sender exists in records:
     - The primary contact and the primary account are updated with the email ID.
   - If the From email ID does not exist in the database:
     - A message is added to the SR that the sender can't be identified.
   - If there is more than one contact in the database with the same email ID:
     - The Primary Contact field on the SR is left blank. A message is added to the SR, stating the reason and also showing the email ID from the original incoming email.
6. If there are other email IDs in the message in the To or CC fields that can be identified, those IDs are added as contacts. If other email IDs can't be identified, the email IDs are ignored.
7. The incoming message content and the attachments are added to the new SR.

Setting Up Email Channels: Procedure

The tasks to set up email channels involves setting up inbound and outbound email channels. The deploying customer can setup either one of the channels, and can also use a single email channel to handle both inbound and outbound emails.

Inbound channels refer to the messages that are received by the customers and outbound messages refer to the messages that are sent to the customer.

Setting up an email channel includes the following steps:

1. Sign in as an administrator to Oracle Engagement Cloud.
2. Select the Setup and Maintenance option.
3. Configure an email channel for your service organization. For more information about this procedure, see Configuring an Email Channel: Procedure.
4. Setup Email Filters. Filters enable you to set one or more criteria based on which an incoming message from a customer or a partner can be accepted or rejected. For more information about setting up filters, see Creating and Updating Inbound Message Filters: Procedure.
5. Configure a Job to process inbound emails to retrieve emails from the customer at regular intervals. For more information, see Configuring a Job to Process Inbound Emails: Procedure.
6. Configure inbound and outbound email profile options. For more information, see Configuring Email Profile Options: Procedure.

After configuring the email channels, inbound and outbound message from customers can be processed.
Inbound Message Filters: Explained

Inbound message filters enable you to set one or more filter criteria based on which an incoming message from a customer or a partner can be accepted or rejected. For example, you can set a filter to reject any mails that have the subject AutoReply. Message filter patterns are specified using regular expressions. For example, to specify a filter pattern to accept all e-mails from Oracle employees, the regular expression is given as ^[A-Za-z0-9._%+-]*@oracle\.com. You can set more than one filter and order them by priority. When a message comes in, the filter criteria are checked in the order of priority. If the first filter criteria does not apply to a message, the subsequent ones are checked. If any of the filter criteria match the incoming message, the message is accepted or rejected based on the filter specifications.

Creating and Updating Inbound Message Filters: Procedure

To view the existing inbound message filters:

1. Sign in as an administrator to Oracle Engagement Cloud.
2. On the Navigator, click **Setup and Maintenance**.
3. Select **Service** from the **Setup** menu.
4. Click the Communication Channels functional area.
5. Select **All Tasks** from the **Show** drop-down list.
6. Click the **Manage E-mail Filters** task.
   
The **Inbound Message Filters** page appears, showing a list of the existing inbound message filters.

Creating a New Message Filter

To create a new inbound message filter:

1. In the **Inbound Message Filters** page, click **Create**.
2. In the **Create Message Filter** page, select a filter type. Filter type indicates the message part on which the filter is applied. A filter type can be one of the following:
   - File attachment: A filter is applied to the attached file type.
   - Header: A filter is applied to the message header.
   - Mime attachment: A filter is applied to the Content-Type header.
   - Reply to: A filter is applied to the Reply To address.
   - Sender: A filter is applied to the sender of the message.
   - Subject: A filter is applied to the subject of the message.
3. Specify a **Field Name**. For File attachment and Mime attachment filters, the field name is not mandatory.
4. Specify an alphanumeric **Filter Pattern**.
5. Select an **Action on Pattern Match** to Accept or Reject.
6. Specify an optional **Description**.
7. Click **Create**.

**Note:** All filters are enabled by default. To disable a filter, update the properties as described in the following section.
Updating Inbound Message Filters

To update an inbound message filter, perform the following steps:

1. In the **Inbound Message Filters** page, click a filter type.
2. In the **Update Message Filter** page, change the filter properties. You can change all of the properties except the filter type.

   **Note:** If you want to disable a filter, select **Yes** in the **Disabled** field. The default value is **No**.

3. After modifying the filter properties, click **Save and Close**.

Inbound Message Filters: Examples

The following table lists examples of regular expressions for filter patterns.

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Regular Expression Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>File attachment</td>
<td>Attachments that are files with either .pdf, .txt, or .html file extensions:</td>
</tr>
<tr>
<td></td>
<td>`^[A-Za-z0-9_\ ]*.pdf</td>
</tr>
<tr>
<td>Header</td>
<td>Emails with headers that contain the string Gentle Reminder:</td>
</tr>
<tr>
<td></td>
<td><code>^Gentle Reminder[A-Za-z0-9. %+-]*</code></td>
</tr>
<tr>
<td>Mime attachment</td>
<td>Content Type header with values such as text/plain, text/html, image/jpeg, or application/octet-stream</td>
</tr>
<tr>
<td></td>
<td>text/plain</td>
</tr>
<tr>
<td>Reply to</td>
<td>Emails that are sent by the support team:</td>
</tr>
<tr>
<td></td>
<td><code>^support_[A-Za-z0-9-_ %+-]+@company\. com</code></td>
</tr>
<tr>
<td>Sender</td>
<td>Emails that are sent by an Oracle employee:</td>
</tr>
<tr>
<td></td>
<td><code>^[A-Za-z0-9-_ %+-]+@oracle\. com</code></td>
</tr>
<tr>
<td>Subject</td>
<td>Email subjects with string AutoReply:</td>
</tr>
<tr>
<td></td>
<td><code>^Auto Reply: [A-Za-z0-9-_ %+-]+</code></td>
</tr>
</tbody>
</table>
Defining Email Templates: Procedure

You can create email templates for Forward, Response, and System Response messages of a service request (SR). You can create templates using HTML to send email notifications for an SR. To define an email template, use the following procedure. For more information, see the Oracle Sales Cloud Extending Sales guide.

1. Create a sample HTML file for email message using any HTML authoring tool. You can use SR fields within the email content. For example, the message can be "This is an update about SR [$SRNumber$] ".
2. Sign in to Oracle Cloud application as an administrator.
3. Navigate to Application Composer.
4. Select Service application.
5. In the Email Templates page, click the plus icon to create a new template.
6. Select Service Request as an Object.
7. Specify a name for the template.
8. Specify a description.
9. To add any attachments, click the plus icon, browse to the file location, and select the file.
10. A template is active by default. To disable the template, clear the Active option.
11. Specify the email subject. You can use SR field names in the subject. For example, subject can be Resolved issue [$Title$].
12. Edit the message HTML as needed. Add #MessageContent# tag anywhere in the HTML code. This tag is replaced by the SR message content.
13. In email templates that are meant for forwarding to internal users, you can include a link to the SR within the template. Include the link in the following format:
   <Link to company's engagement cloud site>/service/faces/FuseOverview?
   fndGlobalItemNodeID=itemNode_service_service_requests&pSrNumber=<SR Number>.
   For example, https://company123.us.oracle.com:10616/service/faces/FuseOverview?
   fndGlobalItemNodeID=itemNode_service_service_requests&pSrNumber=SR0000029093.
14. Click Save and Close.

Related Topics

- E-Mail Templates: Explained

Configuring Email Profile Options: Procedure

Email profile options enable you to set options for incoming and outgoing emails, such as the Reply To email ID that is to be used to send out a receipt to the customer that sent a message. Configure the inbound email profile options as specified in the following table.

<table>
<thead>
<tr>
<th>Inbound Profile Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_INBOUND_REPLYTO_EMAIL</td>
<td>Indicates the Reply To email that is used for the acknowledgment emails that are sent to the customers. This must be a no reply email. For example, Support <a href="mailto:noreply@oracle.com">noreply@oracle.com</a>.</td>
</tr>
<tr>
<td>SVC_ENABLE_INBOUND_EMAIL_DEFAULT_PROCESSING</td>
<td>Indicates if inbound emails from customers must be processed automatically by creating or updating an SR. If this option is disabled, only the inbound message object is created, without creating an SR.</td>
</tr>
</tbody>
</table>
### Inbound Profile Options

<table>
<thead>
<tr>
<th>profiling_option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_INBOUND_EMAIL_ADDRESSES</td>
<td>Indicates the inbound email IDs that are monitored by the Service application. The email IDs are separated by a comma and are automatically populated during provisioning. These inbound email IDs must not be updated by the deploying customer. The customer must use the address to set a forwarding rule.</td>
</tr>
<tr>
<td>SVC_INBOUND_EMAIL_MAX_ATTACHMENT_SIZE</td>
<td>Indicates the maximum size of attachments that are permitted in an inbound email, in MB.</td>
</tr>
<tr>
<td>SVC_INBOUND_MESSAGE_BATCH_SIZE</td>
<td>Indicates the number of emails that can be processed at a given time.</td>
</tr>
<tr>
<td>SVC_EMAIL_PROCESS_UNKNOWN_CUST</td>
<td>Indicates if an SR must be created for emails sent by unknown customers.</td>
</tr>
<tr>
<td>SVC_ENABLE_INBOUND_EMAIL_ACKNOWLEDGEMENT</td>
<td>Indicates if an acknowledgment must be set for an incoming email or not.</td>
</tr>
</tbody>
</table>

Configure the outbound profile options for CRM as specified in the following table.

<table>
<thead>
<tr>
<th>profiling_option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_SR_FORWARD_TEMPLATE_NAME</td>
<td>Indicates the email template name for SR messages of type Forward.</td>
</tr>
<tr>
<td>SVC_SR_RESPONSE_TEMPLATE_NAME</td>
<td>Indicates the email template name for SR messages of type Response.</td>
</tr>
<tr>
<td>SVC_SR_SYSTEM_RESPONSE_TEMPLATE_NAME</td>
<td>Indicates the template name for SR messages of type System Response.</td>
</tr>
<tr>
<td>SVC_SR_EMAIL_ATT_SIZE</td>
<td>Indicates the maximum size of attachments for outbound emails that are sent from the Service application.</td>
</tr>
</tbody>
</table>

Configure the outbound profile options for HCM, as specified in the following table.

<table>
<thead>
<tr>
<th>profiling_option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_SR_FORWARD_TEMPLATE_NAME_HRD</td>
<td>Indicates the email template for the HR Help Desk Service Request Forward messages.</td>
</tr>
<tr>
<td>SVC_SR_RESPONSE_TEMPLATE_NAME_HRD</td>
<td>Indicates the email template for HR Help Desk Service Request Response messages.</td>
</tr>
<tr>
<td>SVC_SR_SYSTEM_RESPONSE_TEMPLATE_NAME_HRD</td>
<td>Indicates the email template for HR Help Desk Service Request System Response messages.</td>
</tr>
</tbody>
</table>
Configuring Profile Options

To configure email profile options, perform the following steps.

1. Navigate to Setup and Maintenance.
2. Click the Setup drop-down list and select Service.
3. Select the Communication Channel functional area.
4. To configure inbound profile options, select the Manage Inbound Email Profile Options task. To configure outbound profile options, select the Manage Outbound Email Profile Options task.
5. Click the name of the profile option to be set.
6. In the Manage Email Profile Options page, in the Profile Values section, click the plus icon to add a value.
7. Click Save.

Setting Resource Name as From Name in Outbound Emails

When you send emails to your customers, the From name in the email is typically the channel name. However, you have the option to set the resource name as the From name in outbound emails by configuring the SVC_USE_RESOURCE_NAME_IN_OUTBOUND profile option.

To set the service agent’s name as the From name in emails sent to customers:

1. Ensure that you have configured the SPF policy to enable outbound email so that your emails are not rejected by your customer's mail server. For more information about configuring the SPF policy, see “Configuring an Email Channel: Procedure”.
2. Sign in as a setup user or administrator.
3. Navigate to the Setup and Maintenance work area and open the panel tab.
4. Click Search from the list of displayed tasks.
5. Search for and select Manage Profile Options.
6. In the Search Results : Profile Options area of the Manage Profile Options page, click New.
7. In the Create Profile Option page, create a profile option by specifying the following values:
   a. Profile Option Code: SVC_USERESOURCE_NAME_IN_OUTBOUND
   b. Profile Display Name: Use resource name as From name in outbound emails.
   c. Application: Service
   d. Module: Service
   e. Description: Specify that the resource name must be used as the From name in emails sent to customers.
   f. SQL Validation: SELECT meaning, lookup_code FROM Fnd_lookups WHERE lookup_type = 'YES_NO' AND enabled_flag = 'Y'
8. Save the record.
   The Manage Profile Options page is displayed.
9. In the Profile Options area, ensure that the row with the SVC_USERESOURCE_NAME_IN_OUTBOUND profile option is selected.
10. In the Profile Option Levels area, select the respective check boxes for the Site Level so that it’s enabled and updatable.
11. Click **Save and Close**.
12. Search for and select **Manage Administrator Profile Values**.
13. Navigate to the Search area of the **Manage Administrator Profile Values** page.
14. In the **Profile Option Code** field, enter `SVC_USE_RESOURCE_NAME_IN_OUTBOUND` and click **Search**.
15. In the Profile Options area, select the row that has the `SVC_USE_RESOURCE_NAME_IN_OUTBOUND` Profile Option Code.
16. In the Profile Values area, click **New**.
17. In the **Profile Level** field, select **Site**.
18. From the **Profile Value** drop-down list for **Site**, select **Yes**.
19. Click **Save**.

### Configuring an Email Channel: Procedure

Set up an email channel to send and receive emails from the customers.

An email channel can be the following.

- **Inbound email**: Indicates the service emails received from your customers. As part of your implementation, you must set up a forwarding rule on your company email server to redirect these emails to the email account that Oracle provided at the time of provisioning or Oracle’s inbound email ID. For example, all the support emails that are sent to TechSupport@company.com are forwarded to `pod_name.fa.intservice.incoming@pod_name-opcwf.mail.dcsn.oraclecloud.com` for processing. The `SVC_INBOUND_EMAIL_ADDRESSES` profile option indicates the Oracle email ID to which the support mails must be forwarded.

  **Note**: The `SVC_INBOUND_EMAIL_ADDRESSES` profile option contains two email IDs. Use the email ID that contains the text ‘Extservice’ as a part of the ID, as the forwarding email ID.

- **Outbound email**: Indicates the emails that are sent by the Service application from the service request; for example, when an agent responds to the primary customer contact. To ensure that your outbound email is delivered successfully to your external recipients, you must set up a Sender Policy Framework (SPF) policy on your domain. To enable Oracle to send out an email on your behalf, the deploying company must
  - Set up an SPF policy on their domain as an authentication mechanism. The exact method of setting up an SPF policy varies from one domain provider to another. For example, `v=spf1 include:spf_c.oracle.com ~all`.
  - Set values for `SVC_USERESOURCE_NAME_IN_OUTBOUND` and `SVC_USERESOURCE_EMAIL_IN_OUTBOUND` profile options to **Yes**. For more information, see Configuring Email Profile Options: Procedure.

To configure an email channel for the customer, perform the following steps:

1. Navigate to **Setup and Maintenance**.
2. Click the **Setup** drop-down list and select **Service**.
3. Select the **Communication Channels** functional area, and then select **All Tasks** in the **Show** drop-down list.
4. Click the **Manage Communication Channels** task.
5. In the **Service Channels** screen, click **Create Channel**.
6. In the **Create Channel** screen:
   - Select a **Stripe Code**. Select **HCM** to process emails from and to internal employees, through the HCM Help desk support. Select **CRM** to process emails from and to external customers.
   - Select the **Channel Type** as **Email**.
c. Specify the deploying company’s support email ID as the **Account Name**. If a forwarding rule is configured, all the mails that are sent to the specified support email ID are forwarded to the Oracle’s inbound email ID. If an outbound email is configured, Oracle can send mails to the customer as the specified support email ID, on behalf of the deploying company.

d. Verify whether the generated **Channel Code** is unique. The channel code is autogenerated and it uniquely identifies a communication channel when exporting or importing channels from one environment to another.
   - If the autogenerated channel code is unique, don't change it.
   - If the autogenerated channel code is not unique, add a set of characters to the code to make it unique.

e. Specify a **Display Name** to indicate any information about the channel, such as the name of the deploying company for which the channel is being configured.

f. The newly created channel is active by default. To deactivate it, clear the **Active** option.

g. Click **Save**.

### Configuring a Job to Process Inbound E-mails: Procedure

Configure a new job to retrieve e-mails at regular intervals, based on the specified frequency. To configure a job process perform the following steps.

1. From the Navigator menu, select the **Scheduled Processes** option.
2. In the **Scheduled Processes** screen, click **Schedule New Process**.
3. In the **Schedule New Process** dialog box, select **Job** as the **Type**.
4. Search for and select the **Retrieve Inbound E-Mail Messages** option from the **Name** drop-down list.
5. Click **OK**.
6. In the **Process Details** screen, Click **Advanced**.
7. In the **Schedule** tab, in the **Run** options, select the **Using a schedule** option.
8. Select **Frequency** and specify a **Start Date**.
9. Click **Submit**.

A job is scheduled.

### Enabling Service Request Outbound E-Mail: Explained

You can enable automatic e-mail notifications for service requests. E-mail notifications are sent when an internal resource posts a response message to a service request. This topic describes how to enable outbound e-mails for service requests.

To enable outbound e-mails for service requests:

1. Sign in as an administrator or a setup user.
2. Navigate to the **Setup and Maintenance** work area and click **Service**.
3. Click the **Actions** drop-down list, then select **View Configuration**.
   The **Configure Service** page appears.
4. For the Service offering, click **Features**.
5. Select the **Service Request Outbound E-mail** check box.
6. Click **Done**.
The ability to send service request outbound e-mails is now enabled.

The next task is to create e-mail templates for responses to customers and for forwarding the service request to others. Once you have created the templates, specify the template names on the Manage Service Request E-mail Message Profile Options task.

Related Topics

- E-Mail Templates: Explained
- Managing Service Request E-Mail Message Profile Options: Explained

Modifying Acknowledgment Messages for Inbound Email

When you receive an email from your customer or employee to open a service request, an acknowledgment message is sent to them automatically. You can modify the predefined acknowledgment messages provided by the application according to your requirement.

If you have deployed CRM, the following predefined messages are provided:

- SVC_EMAIL_ACK_FOR_KNOWN_CUST
- SVC_EMAIL_ACK_FOR_MULTI_CUST
- SVC_EMAIL_ACK_FOR_UNKOWN_CUST

If you have deployed HR Help Desk, the following predefined messages are provided:

- SVC_EMAIL_ACK_FOR_KNOWN_EMP
- SVC_EMAIL_ACK_FOR_MULTI_EMP
- SVC_EMAIL_ACK_FOR_UNKNOWN_EMP

To modify the acknowledgment messages that you send out to your customers or employees when you receive an inbound email:

1. Sign in as a setup user or administrator.
2. Navigate to the Setup and Maintenance work area and open the Tasks panel tab.
3. Click Search from the list of displayed tasks.
4. In the Search window, enter Manage Messages.
5. Click the Manage Messages task that is displayed.
   The Manage Messages window is displayed.
6. In the Message Name field, enter %SVC%ACK%.
7. From the list of acknowledgment messages displayed, select the message that you want to edit and click Edit.
   The Edit Message window is displayed.
8. Under the Message Text area, edit the Short Text and User Details the way you want.
9. Click Save.
10. If you want to translate the modified messages, click Translation Editor.
    The Edit Translations window displays the list of available languages for translating the messages.
11. Select a row and click in the Short Text field.
12. In the Short Text window, edit the message and click OK.
13. Click in the User Details field, edit the text, and click OK.
14. Repeat steps 11 to 13 for each row to edit the messages for all the available languages.
15. Click OK in the Edit Translations window to save the changes.
16. Click Save and Close in the Manage Messages window.

Setting Up Social Channels: Explained

The social channel enables you to reach customers where and when they engage with you on social networks, and provide faster service. This channel enables you to create service requests (SRs) from social network posts and provide customer service through the social network, while tracking them in Oracle Engagement Cloud.

To use the social channel, you must integrate Engagement Cloud with a platform that manages your social network posts, such as Oracle Social Cloud, using Oracle Integration Cloud. Once the integration is complete, you can set up the social channel for SRs.

As an administrator, you can enable the social channel in Engagement Cloud to allow creating SRs based on social network posts. Social posts from the third-party social relationship management software are added as SRs in Engagement Cloud and are assigned to agents.

The following list provides an overview of the steps required to set up social channels in Engagement Cloud.

1. Create social channels for SRs in Engagement Cloud.
2. Integrate Engagement Cloud with Social Cloud.

   See "Integrating Engagement Cloud with Social Cloud: Procedure" for information about creating an integration user, importing the integration package, creating connections, and activating the integrations for the social channel.

Process Flow for Social Posts

The process flow for social posts is as follows:

1. Community managers in Social Cloud send social posts to Engagement Cloud and create SRs for a social post.
2. Support agents review and respond to social posts (both public and private).
   a. Service requests are created in Engagement Cloud for social posts sent from Social Cloud.
   b. Once an SR is created, the Social Cloud post is updated with the SR reference number.
   c. Photos and videos added to social posts are displayed as attachments to service request messages.
   d. Users can drill down on the attachments to view the photos and videos.

   ✍️ Note: To view attachments to private messages, users must sign in to Social Cloud.

3. Any new reply or comment by the customer to an existing conversation is automatically added to the existing SR. These messages are added as SR messages.
4. Once an SR is resolved, the Social Cloud post is updated to indicate that the corresponding SR is resolved.
   a. If an SR is reopened after it is resolved, Social Cloud is also updated with the status change.

Supported Social Networks

Engagement Cloud supports Facebook, Twitter, Instagram, Weibo, and WeChat social network channels.

The following table lists the features supported for the social network monitoring scenarios.
Implementing Service in Engagement Cloud

Chapter 7
Setting Up Communication Channels

<table>
<thead>
<tr>
<th>Social Network Monitoring Scenarios</th>
<th>Facebook</th>
<th>Twitter</th>
<th>Instagram</th>
<th>Weibo</th>
<th>WeChat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor public messages</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Monitor comments and replies on public messages</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Monitor private messages</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Monitor replies to private messages</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note:**
- Photos and videos sent along with private messages on both Twitter and Facebook require you to sign in to the Social Cloud application.
- On Instagram, customers cannot post on the brand company page.
- Social Cloud does not retrieve attachments to a private message on Weibo.
- Social Cloud does not support monitoring of public messages on WeChat.

The following table lists the features supported for the social response scenarios.

<table>
<thead>
<tr>
<th>Social Response Scenarios</th>
<th>Facebook</th>
<th>Twitter</th>
<th>Instagram</th>
<th>Weibo</th>
<th>WeChat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reply to public messages</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reply to comments and public messages</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

The following table lists the features supported for other scenarios.

<table>
<thead>
<tr>
<th>Other Scenarios</th>
<th>Facebook</th>
<th>Twitter</th>
<th>Instagram</th>
<th>Weibo</th>
<th>WeChat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update Social Cloud conversation with SR reference number for a social post</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Update Social Cloud conversation when an SR is resolved</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Other Scenarios | Facebook | Twitter | Instagram | Weibo | WeChat
---|---|---|---|---|---
Update Social Cloud conversation when an SR status changes from Resolved to In Progress | Yes | Yes | Yes | Yes | Yes

### Fields Mapped between Social Cloud and Engagement Cloud

Social posts sent by community managers are created as social SRs (SRs with channel type as Social).

The following table describes the field mapping between Social Cloud and Engagement Cloud.

<table>
<thead>
<tr>
<th>Engagement Cloud Field</th>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Channel Type</strong></td>
<td>Social</td>
<td>All social posts</td>
</tr>
<tr>
<td><strong>Channel</strong></td>
<td>Name of the social network where the customer posted the message (for example, Facebook)</td>
<td>Social Cloud</td>
</tr>
<tr>
<td><strong>SR Title</strong></td>
<td>First 400 characters of the social post content</td>
<td>Social Cloud</td>
</tr>
<tr>
<td><strong>Problem Description</strong></td>
<td>First 1000 characters of the social post content</td>
<td>Social Cloud</td>
</tr>
<tr>
<td><strong>Primary Contact</strong></td>
<td>If a contact point is found for the social post author, the primary contact is set in the SR</td>
<td>Auto populated by Engagement Cloud</td>
</tr>
<tr>
<td><strong>SR Account</strong></td>
<td>The primary account associated with the primary contact of the SR (if exists)</td>
<td>Auto populated by Engagement Cloud</td>
</tr>
<tr>
<td><strong>SR Creation Date</strong></td>
<td>Current date</td>
<td>Auto populated by Engagement Cloud</td>
</tr>
<tr>
<td><strong>Service Request Message</strong></td>
<td>Every message or post in the conversation with the customer is created as an SR message</td>
<td>Social Cloud post content</td>
</tr>
</tbody>
</table>

### Creating Social Channels for Service Requests

To use the Social feature, you must first create the social channel in Engagement Cloud.

To create social channels, do the following:

1. Sign in as an administrator to Engagement Cloud.
2. Navigate to **Setup and Maintenance**.
3. From the **Setup** drop-down list, select **Service**.
4. Select the Communication Channels functional area, and then click Manage Communication Channels. The Service Channels page is displayed.
5. Click Create Channel. The Create Channel dialog box is displayed.
6. Do the following:
   a. From the Stripe Code drop-down list, select CRM.
   b. From the Channel Type drop-down list, select Social.
   c. From the Network drop-down list, select a supported social network.
   d. In the Account Name field, enter the social handle or fan page name of the selected social network. This must match the social resource name in the integrating Social Cloud application.
   e. Verify whether the generated Channel Code is unique.
      - Channel Code is automatically generated by the application. You can use this autogenerated code as is, and you must modify it only when it is not unique. To make it unique, you can add any set of characters.
      - Channel Code is used by the application to uniquely identify a channel when exporting and importing channels from one environment to another.
   f. Enter a Display Name to provide information about the channel, such as the name of the deploying company for which the channel is being configured.
   g. From the Business Unit drop-down list, search for and select a business unit, when multiple business units are configured.
   h. To deactivate the newly created channel, clear the Active option. It is active by default.
   i. Click Save.
7. Create a new channel for every social network that you want to use.
8. Save the social channels.

Integrating Engagement Cloud with Social Cloud: Procedure

After you have created a social channel in Oracle Engagement Cloud, to use the social channel, you must integrate Engagement Cloud with Oracle Social Cloud.

To integrate Engagement Cloud with Social Cloud, do the following:

1. Create an integration user with privileges to perform the integration with Social Cloud.
2. Import the integration package from Oracle Marketplace.
3. Import SSL certificates for Engagement Cloud and Social Cloud.
5. Activate the integrations.
7. Set up CSF SOA key after activating the integrations.

Create an Integration User for the Social Channel

To integrate Engagement Cloud with Social Cloud, Oracle recommends that you create a user specifically for the integration. The integration user can call the Sales Cloud service catalog or event catalog web services from Integration Cloud.

The following procedure describes how to create an integration user and what privileges to provide to the role.

1. Sign in to Oracle Sales Cloud as an administrator.
2. Select Navigator > My Team > Users, Roles and Delegations, and click Manage Users.
3. On the Manage Users page, select Create New User from the Actions menu.
4. Enter the following details for the new user:
   - **Last Name**: Enter a last name for the user.
   - **E-mail**: Enter a valid email ID for the user.
   - **Hire Date**: Select today’s date.
   - **User Name**: Enter a user name for the user.
   - **Person Type**: Select Employee from the drop-down list.
   - **Legal Employer**: Select the legal employer from the list.
   - **Business Unit**: Select a valid business unit.
   - **Send user name and password**: Select this option.
   - **User Log in**: Enter the user name you created.
   - **Password**: Enter the password for the user.
5. Save the user details. An email is sent to the address after the user has been created.
6. Check the user credentials sent in the email, sign in as the new user and reset the password. After creating the user, sign in to the security console and provide the following roles to the integration user
   - **SOA Operator**
     The SOA Operator is a duty role and duty role cannot be directly assigned to a login using security console. Create an enterprise role as a parent of SOA Operator and associate that enterprise role to the integration user.
   - **Customer Service Representative**
   - **Resource**

**Import the Integration Package**

After you have set up Engagement Cloud and created the integration user with the required privileges, you can set up the Integration Cloud integration package.

Before you start setting up Integration Cloud, go to the Oracle Marketplace and search and download the Engagement Cloud to Social Cloud integration package. After you download this package, perform the following procedures to set up Integration Cloud. For more information about Oracle Marketplace, see the Oracle Cloud Marketplace documentation.

The Integration Cloud integration package supports the following

- **Inbound**
  - Create an SR in Engagement Cloud for a social post.
  - Add a social post comment as a message to an SR.
- **Outbound**
  - Respond to a customer on the social network that they posted on.
Import SSL Certificates
You must import the SSL certificates before you configure and activate the connections.


Activate the Connections to Engagement Cloud and Social Cloud
After you download the package and import the SSL certificates, connect to Engagement Cloud using the Sales Cloud adapter. The procedure for importing the adapter is described in the topic "Creating an Oracle Sales Cloud Adapter Connection", available at https://docs.oracle.com/en/cloud/paas/integration-cloud-service/icssc/index.html.

The following procedure describes how to configure the connections to the Sales Cloud instance.

1. Sign in to the Integration Cloud application using your integration user credentials.
2. Click the Connections icon on the home page.
3. Ensure that Oracle Engagement Cloud is listed on the connections page.
4. Click Oracle Engagement Cloud to view the details.
5. Click Configure Connectivity and enter the following information in the Connection Properties dialog box:
   - OSC Service Catalog WSDL URL: Enter the service catalog URL on your Engagement Cloud instance
   - (Optional) OSC Event Catalog URL: Enter the event catalog URL on your Engagement Cloud instance
6. Click OK.
7. Click Configure Security and enter the following information:
   - Security Policy: Enter Username Password Token.
   - User name: Enter the integration user name.
   - Password: Enter the password for the integration user.
   - Confirm Password: Reenter the password.
8. Click OK.
9. Click Test on the Actions bar of the Sales Cloud page.
10. Ensure the connection test is successful and the status meter shows 100%.
11. Click OK.

The following procedure describes how to configure the Oracle Social Cloud connection.

1. Sign in to the Integration Cloud application using your integration user credentials.
2. Select Connections and then create a new connection for Oracle Social Cloud.
3. Enter the connection details as shown in the following table:

<table>
<thead>
<tr>
<th>Section</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Properties</td>
<td>- Connection Type: REST API Base URL</td>
</tr>
<tr>
<td></td>
<td>- TLS Version: Not Applicable</td>
</tr>
<tr>
<td></td>
<td>- Connection URL: <a href="https://srn-api-temp.slc05hpo.oracle.com">https://srn-api-temp.slc05hpo.oracle.com</a></td>
</tr>
</tbody>
</table>
Section | Details
--- | ---
Authorization Request | Enter the authorization request URL. The URL must have the following parameters defined:
  - scope=engage
  - response_type=code
  - redirect_uri=${redirect_uri}
  - To get your client ID, see Register your Client Application for procedure. For example, a sample authorization request URL looks like this: `https://gatekeeper.srm-integration.pp1.oraclecloud.com/oauth/authorize?scope=engage&response_type=code&redirect_uri=${redirect_uri}&client_id=a9a5fb2b482545172fd6101e3c16cc1e9ba140742aa630b6c3e9055a89e6e9a2`

Access Token Request | Enter the access token request with the correct URL, client ID, and client secret. For example, a sample access token request looks like this: `-X POST -H 'Content-Type: application/x-www-form-urlencoded' -d 'false' 'https://gatekeeper.srm-staging.pp1.oraclecloud.com/oauth/token?code=${auth_code}&client_id=a9a5fb2b482545172fd6101e3c16cc1e9ba140742aa630b6c3e9055a89e6e9a2&client_secret=de969db4808a87746dbaf1fa648d552aa6dd1927da5ed2ee731c7ad62ee61b19&grant_type=authorization_code'`

Refresh Token Request | Enter the refresh token request with the URL, refresh token, client ID, and client secret. For example, a sample access token request looks like this: `-X POST -H 'Content-Type: application/x-www-form-urlencoded' -d 'false' 'https://gatekeeper.srm-staging.pp1.oraclecloud.com/oauth/token?refresh_token=${refresh_token}&client_id=a9a5fb2b482545172fd6101e3c16cc1e9ba140742aa630b6c3e9055a89e6e9a2&client_secret=de969db4808a87746dbaf1fa648d552aa6dd1927da5ed2ee731c7ad62ee61b19&grant_type=refresh_token'`

4. Save the configuration and test the connection.

Activating the Integrations

You must activate the following integrations.

- Social Cloud Conversation to Engagement Cloud: Creates Engagement Cloud social post records from Social Cloud conversation record and its related posts by mapping the object attributes of the two applications. Engagement Cloud further processes the social posts to create service request and service request messages.

- Engagement Cloud Service Request to Social Cloud: Updates the Social Cloud conversation object with the service request details.

- Engagement Cloud Reply to Social Cloud: Updates the Social Cloud conversation with the response from Engagement Cloud user to the social customer. Social Cloud in turn sends the response to the customer on the social network.

- Engagement Cloud Update SR to Social Cloud: Updates Social Cloud conversation with the Engagement Cloud service request number.

- Engagement Cloud Sync SR to Social Cloud: Updates the Social Cloud conversation with the service request reference number from Engagement Cloud. This integration is called when an Engagement Cloud user sends a response.

After you configure the connections, activate the integrations as follows:

1. Sign in to the Integration Cloud application using your integration user credentials.
2. Click the **Integrations** icon on the home page.
3. Search for each integration by its name.
4. Click the **Activate** button on each integration.
5. Select the **Enable detailed tracing** option, and click **Activate** on the confirmation dialog box.
6. Ensure the flow has been activated successfully.
7. Repeat the previous steps for each integration.
Setting Up the Integration Cloud Plug-in in Social Cloud

Enter CSF Key through SOA Composer
After activating the connections, configure the CSF key through the SOA Composer.

The following procedure describes how to configure the CSF key:

1. Sign in to the SOA Composer at https://host:port/soa/composer as a user with SOA_Admin role.
2. Click Manage Security.
3. In the Manage Credentials group, enter the following information:

   - **csf-key**: The Integration Cloud subscription identity domain. This value is the domain name of the Integration Cloud instance that has been provisioned for your integration and is provided in the sign-in email that you received from the Integration Cloud subscription.
   - **User Name**: The user name of the user who is granted access to Integration Cloud.
   - **Password**: The password credentials for the user who is granted access to Integration Cloud. Confirm the password value that you provide.

Engagement Cloud to Social Cloud Integration Mapping: Explained
This topic describes the mapping between Oracle Engagement Cloud and Oracle Social Cloud through Integration Cloud.

Social Cloud Conversation to Engagement Cloud
This section describes the integration mappings for the Social Cloud Conversation to Engagement Cloud integration.

**ProcessSocialUser**
ProcessSocialUser maps the social network user to a social user in Engagement Cloud. The identified user is used in the **ProcessSocialPost** mapping. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributions. Content. Author.Name</td>
<td>String</td>
<td>SocialNetworkUserName</td>
<td>String</td>
<td>User’s name as shown on the social network.</td>
</tr>
</tbody>
</table>
### Social Cloud Attribute | Data Type | Engagement Cloud Attribute | Data Type | Description
--- | --- | --- | --- | ---
Contributions. Content. Author. AuthorImage | String | SocialNetworkUserImage | String | URL of the social user’s profile image.
Contributions. Content. Author. AuthorProfileURL | String | ProfileURL | String | Profile URL of the social user.
Contributions. Content. Resource. ResourceName | String | SocialNetworkCd | String | Social network, such as Twitter and Facebook.

**ProcessSocialPost**

ProcessSocialPost maps the social post in Social Cloud to the social post in Engagement Cloud. The following table describes the integration mapping.

> **Note**: For Social Cloud attributes related to contributions, the mapping applies to all contributions (posts) within a conversation.

### Social Cloud Attribute | Data Type | Engagement Cloud Attribute | Data Type | Description
--- | --- | --- | --- | ---
QueryParameters. TransactionID | String | SourcePostIDTertiary | String | Transaction identifier of the social post.
ID | Integer | SourcePostIDPrimary | String | Identifier of the post in Social Cloud.
PrimaryContribution. ID | Integer | PostParentPostID | String | Identifier of the social conversation’s primary post.
Contributions. ID | Integer | PostPostID | String | The social post's identifier on the social network.
Contributions. Content. ExternalLink | Integer | PostURL | String | URL of the social post.
<table>
<thead>
<tr>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributions. Content. Author.Name</td>
<td>String</td>
<td>PostUser</td>
<td>String</td>
<td>The social post author’s handle on the social network.</td>
</tr>
<tr>
<td>Contributions. Content. PostedAt</td>
<td>String</td>
<td>PostDate</td>
<td>DateTime-Timestamp</td>
<td>Date of the social post.</td>
</tr>
<tr>
<td>Contributions. Content. Body</td>
<td>String</td>
<td>PostContent</td>
<td>Base64Binary-DataHandler</td>
<td>Content of the social post.</td>
</tr>
<tr>
<td>Contributions. Content. Labels</td>
<td>String</td>
<td>SocialPostTags. Tag</td>
<td>String</td>
<td>Tag associated with the social post.</td>
</tr>
<tr>
<td>Contributions. Content. Attachments. URL</td>
<td>String</td>
<td>SocialPostURLs. URL</td>
<td>String</td>
<td>URL of the attachments in the social post. For most social networks, attachments are either photos or videos.</td>
</tr>
<tr>
<td>Contributions. Content. Resource. ResourceName</td>
<td>String</td>
<td>PostSubChannelName</td>
<td>String</td>
<td>The channel on which the social post was made. For example, the Facebook fan page.</td>
</tr>
<tr>
<td>Contributions. Content. Resource. ResourceType</td>
<td>String</td>
<td>PostChannelCd</td>
<td>String</td>
<td>The social network on which the social post was made.</td>
</tr>
</tbody>
</table>

*Note:* A social post may be associated with multiple tags. Tags are known as Labels in Social Cloud.

*Note:* A social post may be associated with multiple attachments.
UpdateContributions

UpdateContributions provides mapping for the return object to Social Cloud to indicate that a social post was received by Engagement Cloud. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Integer</td>
<td>TemplateParameters. ID</td>
<td>String</td>
<td>Identifier of the Social Cloud conversation.</td>
</tr>
</tbody>
</table>

Engagement Cloud Service Request to Social Cloud

This section describes the integration mappings for the Engagement Cloud Service Request to Social Cloud integration.

UpdateConversation

UpdateConversation provides mapping to update Social Cloud with the service request reference number and status. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FindSRSocialPostResponse. Result. SRNumber</td>
<td>String</td>
<td>ExternalID</td>
<td>String</td>
<td>Reference number of the service request created for the social post.</td>
</tr>
</tbody>
</table>
Engagement Cloud Update SR to Social Cloud
This section describes the integration mappings for the Engagement Cloud Update SR to Social Cloud integration.

**UpdateConversation**

UpdateConversation provides mapping to update Social Cloud whenever there is a status change in the social service request. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnEvent.</td>
<td>String</td>
<td>TopLevelArray. Value.</td>
<td>String</td>
<td>Reference number of the service request created for the social post.</td>
</tr>
<tr>
<td>SRSelectedAttributeChanged</td>
<td>String</td>
<td>TopLevelArray. Value.</td>
<td>String</td>
<td>Status of the service request.</td>
</tr>
<tr>
<td>ExternalID</td>
<td>String</td>
<td>BundleID</td>
<td>String</td>
<td>Bundle identifier of the post in Social Cloud.</td>
</tr>
</tbody>
</table>

Engagement Cloud Reply to Social Cloud
This section describes the integration mappings for the Engagement Cloud Reply to Social Cloud integration.

**AddContribution**

AddContribution provides mapping to create a new contribution in the Social Cloud conversation, when a service request message of type **Response** is added in Engagement Cloud. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnEvent.</td>
<td>Long</td>
<td>Request-Wrapper. Value.</td>
<td>String</td>
<td>Identifier of the service request message created for a social post.</td>
</tr>
</tbody>
</table>
### Setting Up Communication Channels

<table>
<thead>
<tr>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
</table>

#### Note:
A service request message is created for every social post that is part of a conversation. Whereas a service request is only created for the primary post in that conversation.

#### ReportProcessingSuccess

This mapping displays the service request message ID of the response composed by the Engagement Cloud user, which was successfully sent to the customer on the social network. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Engagement Cloud Attribute</th>
<th>Data Type</th>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
</table>

#### ReportProcessingError
This mapping displays the service request message ID of the response composed by the Engagement Cloud user, which could not be sent to the customer on the social network. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Engagement Cloud Attribute</th>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnEvent.</td>
<td>MergeSRMessage.</td>
<td>Long</td>
<td>The service request message identifier of the response composed by the Engagement Cloud user.</td>
</tr>
<tr>
<td>GetSRMessageResponse.</td>
<td>SRMessage.</td>
<td>Long</td>
<td></td>
</tr>
<tr>
<td>Result. MessageID</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The error statuses displayed on the service request messages user interface are the Engagement Cloud users’ messages that could not be sent to the customer.

### Engagement Cloud Sync SR to Social Cloud

This section describes the integration mappings for the Engagement Cloud Sync SR to Social Cloud integration.

**UpdateConversation**

This mapping sends the service request details and the Engagement Cloud users’ responses to Social Cloud, to the customer. The following table describes the integration mapping.

<table>
<thead>
<tr>
<th>Engagement Cloud Attribute</th>
<th>Social Cloud Attribute</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSRMessageSocialPostRe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sult. SourcePostIDPrimary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetServiceRequest.</td>
<td>Request-Wrapper.</td>
<td>String</td>
<td>Reference number of the service request created for the social post.</td>
</tr>
<tr>
<td>GetServiceRequestResponse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Result. SRNumber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetServiceRequest.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetServiceRequestResponse</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Result. StatusTypeCd</td>
<td></td>
<td></td>
<td>Status of the service request.</td>
</tr>
</tbody>
</table>
Configuring the Cross-Channel Interactions: Procedure

This topic describes how to set up and configure cross-channel interactions.

Cross-Channel Interactions: Explained

Cross-channel interactions enables the administrators to track and manage inbound and outbound interactions that cross from one channel to another. For example, when a customer begins an inbound-chat communication with an agent, and then the agent promotes that communication to a phone call, the communication throughout its life cycle and across both channels can be tracked and managed.

The administrator can also identify and associate the business objects that have been edited or created during this communication, and configure the business objects that are available for association with the interaction model. The administrator can also turn on cross channel interaction tracking for service requests.

Managing Setup and Configuration for Cross-Channel Interactions

This page provides the administrator with a single page to associate business objects, setup all the associated profile options pertaining to cross-channel interactions, and set the date and time periods that qualify objects for automatic association to an interaction.

To set up and configure for cross-channel interactions, do the following:

1. Sign in to Oracle Cloud application as a Service Administrator or a Service manager.
2. Navigate to Setup and Maintenance.
3. Click the Tasks tab icon, then select Search.
4. Search for the following string:

   Manage Setup and Configuration for Cross-Channel Interactions

5. Click the Manage Setup and Configuration for Cross-Channel Interactions task.
6. The Enable Automatic Association of Objects with Interactions field is selected as No by default. This enables automatic association of the objects with interactions. When you select No, business objects are not associated with interactions automatically.
7. Select the period, in minutes, to go back from the current time while finding a potential interaction based on the creation date in the Search Retrieval Range Prior to the Creation Time field. The default time is set to 5 minutes. You can set the time between 1 minute and 120 minutes.

   Note: If you have selected to not automatically associate business objects with interactions, setting the time here is irrelevant.

8. Select the period, in minutes, to go back from the current time while finding a potential interaction based on last updated date in the Search Retrieval Range Prior to the Last Update Time field. The default time is set to 5 minutes. You can set the time between 1 minute and 120 minutes.

   Note: If the Enable automatic association option is set to No, setting the time here is irrelevant.
9. In the **Automatically Associated Objects** section, you can configure whether to automatically associate service requests with interactions. These are set to Yes by default.

10. Click **Save** or **Save and Close**.

Cross-Channel Options: Explained

The following table describes the type of cross-channel options that you can configure.

<table>
<thead>
<tr>
<th>Cross-channel Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_INTERACTION_AUTO_ASSOCIATION_ENABLE</td>
<td>Enables automatic association of objects to an interaction. It is set to Off by default.</td>
</tr>
<tr>
<td>SVC_INTERACTION_CREATION_DATE_INTERVAL_FOR_AUTO_ASSOCIATION</td>
<td>Sets the period, in minutes, to go back from the current time while finding a potential interaction based on creation date</td>
</tr>
<tr>
<td>SVC_INTERACTION_LAST_UPDATE_DATE_INTERVAL_FOR_AUTO_ASSOCIATION</td>
<td>Sets the period, in minutes, to go back from the current time while finding a potential interaction based on last updated date.</td>
</tr>
<tr>
<td>SVC_INTERACTION_ENABLED_OBJECTS - (for Service Request)</td>
<td>For Service requests, set this to the value SVC_SERVICE_REQUEST.</td>
</tr>
</tbody>
</table>

Using Groovy Scripts for Social Channel: Explained

Groovy is a standard, dynamic scripting language for the Java platform. You write Groovy scripts using Application Composer’s expression builder, which appears in many places as you modify existing objects or create objects. This topic explains how you can use Groovy scripts for social messages and provides samples of Groovy code.

Validating Agent's Response Character Count

This script validates whether an agent’s response for the Twitter channel is less than or equal to 140 characters.

You must add this script as a validation rule to the Message object (child of Service Request object). You must also add a validation message that is displayed to the agents when their Twitter response is greater than 140 characters.

Use the following code as an example:

```java
if(ChannelId == null){
    return true;
}
def channelVO = newView('ChannelVO')
channelVO.appendViewCriteria("ChannelId = '${ChannelId}'")
channelVO.executeQuery()
def networkType = null
if(channelVO.hasNext())
    if(ChannelId == null){
        return true;
    }
def channelVO = newView('ChannelVO')
channelVO.appendViewCriteria("ChannelId = '${ChannelId}'")
channelVO.executeQuery()
def networkType = null
if(channelVO.hasNext())
    if(ChannelId == null){
        return true;
    }
```
This script validates whether the agent’s response begins with the @mention of the customer’s Twitter handle.

You must add this script as a validation rule to the Message object (child of Service Request object). You must also add a validation message that is displayed to the agents when their Twitter response is greater than 140 characters.

Use the following code as an example:

//Validation logic
//For channel other than social ignore
if(ChannelTypeCd != 'ORA_SVC_SOCIAL')
    return true
//For draft ignore
if(StatusCd == 'ORA_SVC_DRAFT')
    return true
//For message other than response ignore
if(MessageTypeCd != 'ORA_SVC_RESPONSE')
    return true
//Get the last social customer entry
def vo = newView('SrMessageVO')
def vc = newViewCriteria(vo)
def vcr = vc.createRow()
def vci1 = vcr.ensureCriteriaItem('MessageTypeCd')
vci1.setOperator('=')
vci1.setValue('ORA_SVC_CUSTOMER_ENTRY')
def vci2 = vcr.ensureCriteriaItem('ChannelTypeCd')
vci2.setOperator('=')
vci2.setValue('ORA_SVC_SOCIAL')
def vci3 = vcr.ensureCriteriaItem('SrId')
vci3.setOperator('=')
vci3.setValue(ServiceRequest?.SrId)
vc.insertRow(vcr)
vo.appendViewCriteria(vc)
vo.executeQuery()
def lastCustEntry=vo.first()
//Get the channel via
def channelViaList=lastCustEntry.channelCommunication
def channelVia = channelViaList.first()
// Using inbound object id as post id, find the social post
def postId = channelVia.InboundObjectId;
def key = key(postId);
def socialPostVO = newView('SocialPostVO')
def socialPosts = socialPostVO.findByKey(key, 1)
// do nothing if the social post is not found
if (socialPosts == null || socialPosts.size() == 0) {
    return true
}
// use the first social post
def socialPost = socialPosts[0];
def networkType = socialPost.PostChannelCd
//do nothing if other than twitter
if(networkType != 'TWITTER') {
    return true
}
//If message content is not having twitter handle as prefix then error out.
def twitterHandle = '@' + socialPost.PostUser + '
if(MessageContent!= null && !startsWith(MessageContent.toString(),twitterHandle)) {
    return false
}
return true;

Applying Customer's Twitter Handle by Default

When an agent composes a response for a Twitter channel service request, this script applies the customer’s Twitter handle by default, in the SR message.

You must add this script as an object level, after create trigger to the Message object (child of the Service Request object).

Use the following code as an example:

//Defaulting logic for twitter response
//If not social then do nothing
if(ChannelTypeCd !='ORA_SVC_SOCIAL')
    return
//If not of type response do nothing
if(MessageTypeCd != 'ORA_SVC_RESPONSE')
    return
//Find last social customer entry
def vo = newView('SrMessageVO')
def vc = newViewCriteria(vo)
def vcr = vc.createRow()
def vc1 = vcr.ensureCriteriaItem('MessageTypeCd')
vc1.setOperator('=')
vc1.setValue('ORA_SVC_CUSTOMER_ENTRY')
def vc2 = vcr.ensureCriteriaItem('ChannelTypeCd')
vc2.setOperator('=')
vc2.setValue('ORA_SVC_SOCIAL')
def vc3 = vcr.ensureCriteriaItem('SrId')
v3c3.setOperator('=')
vc3.setValue(ServiceRequest?.SrId)
vc.insertRow(vcr)
vo.appendViewCriteria(vc)
vo.executeQuery()
def lastCustEntry=vo.first()
def channelViaList=lastCustEntry.channelCommunication
def channelVia = channelViaList.first()
// Using inbound object id as post id, find the social post
def postId = channelVia.InboundObjectId;
def key = key(postId);
def socialPostVO = newView('SocialPostVO')
deSocialPosts = socialPostVO.findByKey(key, 1)
// Do nothing if the social post is not found
if (socialPosts == null || socialPosts.size() == 0) {
    return
}
// Use the first social post
def socialPost = socialPosts[0];
def networkType = socialPost.PostChannelCd
// Do nothing if not twitter
if (networkType != 'TWITTER') {
    return
}
// If twitter then set the MessageContent with twitter handle
def twitterHandle = '@' + socialPost.PostUser + ' ' if (null != twitterHandle)
setAttribute('MessageContent', twitterHandle)

Mapping Social Post Tags

Tags associated with social posts in Social Cloud can be sent to Engagement Cloud. These tags are not displayed in the application, but can be used in Groovy scripts. This script maps social post labels in Social Cloud to social post tags in Engagement Cloud. Agents can use these tags to take appropriate actions on the SR.

For example, use the following code to set the SR severity to SEV1 when the social post is tagged as urgent:

// do nothing if there are no channel vias
if (!channelCommunication.hasNext()) {
    return;
}

// use the first channel via
def channelVia = channelCommunication.next();

// do nothing if the channel via is not a social channel type
if (channelVia.ChannelTypeCd != 'ORA_SVC_SOCIAL') {
    return
}

// using inbound object id as post id, find the social post
def postId = channelVia.InboundObjectId;
def key = key(postId);
def socialPostVO = newView('SocialPostVO');
deSocialPosts = socialPostVO.findByKey(key, 1);

// do nothing if the social post is not found
if (socialPosts == null || socialPosts.size() == 0) {
    return
}

// use the first social post
def socialPost = socialPosts[0];

// iterate over tags on the social post
def socialPostTags = socialPost.SocialPostTags;
while (socialPostTags.hasNext()) {
    def tag = socialPostTags.next();
    // One example: Set the SR Severity to SEV1 if social post is tagged with "urgent"
Setting Chat Profile Options: Procedure

Chat profile options enable you to configure assignment and routing options for chat requests. The following table lists the chat profile options.

<table>
<thead>
<tr>
<th>Profile Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVC_CHAT_OMNI_INTEGRATION</td>
<td>Enables integration of chat with Omnichannel for automatic routing of chat requests to agents</td>
</tr>
<tr>
<td>SVC_CHAT_OFFER_NOTIFICATION_TIMEOUT_VALUE</td>
<td>Enables you to specify the number of seconds that lapse before the chat notification is automatically closed.</td>
</tr>
<tr>
<td>SVC_ENABLE_CHAT</td>
<td>Enables chat interactions</td>
</tr>
</tbody>
</table>

Configuring Chat Profile Options

Set the profile options to enable various chat features. To configure the profile options, perform the following steps.

1. Sign in to the application as an administrator.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Communication Channels functional area and then select Manage Chat Profile Options.
5. To set the profile options, click an option.
6. In the Profile Values section, in the Profile Level drop-down list, ensure that Site is selected.
7. Select Yes from the Profile Value drop-down list.
8. Click Save and Close.

Enabling Chat

To enable the service representatives to use the chat function and to receive chat notifications, you must enable Omnichannel and desktop notifications. For more information about enabling Omnichannel and desktop notifications, see "Enabling Omnichannel: Procedure" in the Implementing Service for Engagement Cloud guide.

The chat function enables a customer to connect with a service representative. The customer initiates the chat from the Digital Customer Service, and a notification is sent to the available representatives in Engagement Cloud.

Note: If you are using the Digital Customer Service application as your consumer application, ensure that a user exists with the role ORA_SVC_CUSTOMER_SELF_SERVICE_USER_ABSTRACT and the privilege SVC_REQUEST_FOR_CHAT. For more information about the security roles, see About Digital Customer Service Roles: Explained.

Related Topics
- Enabling Omnichannel: Procedure
FAQs About Communication Channels

What happens when a customer sends a service email?

If a customer sends an email, the email IDs in the From field is validated against the customer record in the database. If a match is found, and the email is for a new service, then a new SR is created. If the email is regarding an existing service request, then the email content is added as a message to the relevant SR.

If a match to the email ID is not found in the database, then the following applies:

- The value for the SVC_EMAIL_PROCESS_UNKNOWN_CUST profile option is selected. This option specifies how to process an incoming email from unknown customers.
- If the profile option is set to Y, a new SR is created. However, a message is sent to customers indicating that they can't be identified and must provide valid information for further processing of the SR.
- If the profile option is set to N, no SR is created.

Note: All the recipients of the incoming email, including the unknown contacts are listed in the SR message tab.
8 Configuring Computer Telephony Integration (CTI)

Computer Telephony Integration (CTI): Explained

Computer telephony integration (CTI) enables integration of third party media toolbars with Oracle Cloud Applications. Media toolbar is displayed if the company has enabled the partner CTI service, and the signed-in user has the access privileges to a toolbar.

CTI integration provides the following features:
- Notifications of incoming calls and ability to accept or reject the call
- Automatic caller identification
- Ability to search for a contact
- Optional caller verification
- Display of administrator-defined screen pop with caller or service details
- Automatic interaction recording
- Optional call wrap up

Configuring the Media Toolbar: Procedure

An administrator can configure both horizontal toolbar and vertical media toolbars. The media toolbar is displayed:
- If the company has enabled partner Computer Telephony Integration (CTI) service
- When the signed-in user has the access privileges to a toolbar

An administrator can configure multiple media toolbars, and specify a default.

Live Window is an alternative to the media toolbar. For more information about the live window, refer to Configuring Live Window for CTI.

Incoming Call Notifications

While configuring toolbars, you can also configure incoming call notification window. Notification window displays basic details about the incoming call.

Configuring the Media Toolbar

To configure the media toolbar, do the following:

1. Sign in to Oracle Cloud application as a Service Administrator or a Service manager.
2. Navigate to Setup and Maintenance.
3. Click the Tasks tab icon, then select Search.
4. Search for the following string:

Manage Media Toolbar Configuration

5. Click the Manage Media Toolbar Configuration task.

6. In the Manage Media Toolbar Configuration page, click the add icon to add a toolbar.

7. Select Enabled to make the toolbar active. If a default toolbar is already specified, this field is Disabled by default.

8. The Layout is selected as Embedded (Horizontal) by default.

9. Enter a toolbar name.

10. To configure a horizontal toolbar:

a. Enter the URL of the toolbar. This URL is derived from the third-party application that you have installed to integrate the toolbar.

b. To view the toolbar as it appears after configuration, click Preview.

c. Enter a toolbar height that is not more than 70 pixels.

11. You can optionally add a vertical toolbar. To add a vertical toolbar:

a. Select On.

b. Enter a URL based on the third-party toolbar application.

c. Enter a height and a width that are not more than 470 pixels each.

12. To display the incoming call notification:

a. Select On.

b. Enter the URL for the notification window.

c. Enter a window height of maximum 180 pixels and a width of not more than 350 pixels.

13. Select the newly added toolbar from the list of toolbars and click Default to set the toolbar as a default one for all the enabled users.

14. Click Save or Save and Close.

Configuring Live Window and Companion Tab for CTI

This topic describes how to configure Live Window and Companion Tab for CTI.

About Live Window and Companion Tab

Live Window is a user interface in a dedicated window that assists agents in handling interactions with customers using telephony and real-time web channels including Chat, and Live Video. When enabled, you can open Live Window through a dedicated icon under the Omnichannel Headset availability group icon. The media toolbar is an alternative to Live Window. For more information about the media toolbar, refer to Configuring the Media Toolbar: Procedure.

Live Window opens in a separate browser window, which enables agents the flexibility of sizing and placement of the window, based on their environment and needs. Third-party telephony partners have the ability to embed their control toolbars into the Live Window: this enables agents to use all of the control functionality from the telephony partners, such as accept, reject and transfer phone communications.

The Companion Tab is a screen that appears on the Live Window. The tab provides additional screen real estate, enabling you to embed value-added functionality as part of your phone controls. You can use the tab for features such as administrator dashboards or phone reports which provide additional value and flexibility in the engagement center.

Using the ease and flexibility of both the Live Window and Companion Tab, Omnichannel users can use this as a single location for their channel controls, resizing the Live Window if using a single monitor to fit neatly next to their CRM window.
This enables easy movement between customer interactions with access to their CRM transactional information. For users that have two monitors, the Live Window can be moved to the second monitor providing additional real estate for the CRM window.

Configuring Live Window and Companion Tab

To configure Live Window and Companion Tab:

1. Sign in to Oracle Engagement Cloud as a Service Administrator or a Service Manager.
2. Navigate to Setup and Maintenance.
3. Click the Tasks tab icon, then select Search.
4. Search for the following string: Manage Media Toolbar Configuration
5. In the Manage Media Toolbar Configuration page, click the add icon to add a toolbar.
6. Enter a toolbar name.
7. Select Enabled to make the toolbar active. If a default toolbar is already specified, the new toolbar you are creating is Disabled by default.
8. The Layout is selected as Embedded (Horizontal) by default. Select Live Window (Vertical).
9. Specify the communication panel URL in the Communication Panel URL text box.

Note: This URL is derived from the third-party telephony partner application that you have installed to integrate the toolbar.

10. Select Yes in the Companion Tab field to display the companion Tab in the Live Window.
11. Enter the Label to appear on the Companion Tab.
12. Enter an optional Communication Tab Default URL that was provided by your telephony partner. This content is displayed on initialization of the Live Window.
13. To enable Live Window notification, select Enabled for the Live Window Notification option.
14. (Optional) To make Live Window the default toolbar, select the check mark icon in the Default column of the Live Window row.
15. Click Save or Save and Close.

Configuring Business Objects: Procedure

Business object configuration enables you to specify the business objects that can be used for reverse lookup and for screen pop. You can configure a standard business object or a user business object. A standard object is based on an out-of-the-box business object and a user business object is based on a user-defined business object that is created using the Application Composer. For more information about creating user business objects using Application Composer, see Oracle Sales Cloud Extending Sales guide.

To configure a business object:

1. Navigate to Setup and Maintenance.
2. Click the Setup drop-down list and select Service.
3. Select the Communication Channels functional area, and then select Manage Business Objects Configuration task.

The Manage Business Objects Configuration page is displayed. On this page, you can view the System Objects that are read only, and user objects that can be created, modified, and deleted.
Creating a Standard User Business Object

A standard business object is based on a preconfigured business object, such as Service or Queue. To create a standard business object:

1. Click the add icon.
2. Select object type as **Standard**.
3. Type a preconfigured object name. For example, type **ServiceRequest**.
4. Specify the full object path.

   **Note:** The object path is the **Page Path** for the task flow. In commonly used pages, these are pre-seeded. However, if you create a new object, you must contact support to obtain your object path.
5. Click **Add**.
6. Click **Save** or **Save and Close**.

Creating a User Business Object

To create a user business object, you must first create a user object in the application by using the Application Composer. Create the fields, pages, and security settings for the new object. For example, create an object called **SRTickets** in the Service application. For more information about creating a user business object, see Oracle Sales Cloud Extending Sales guide. To create a user business object:

1. Click the add icon.
2. Select object type as **User-Defined**.
3. Select the application in which you created an object. For example, select **Service** application.
4. Type the name of the user-defined object that you created and click **Validate**. For example, type **SRTickets**.
   
   If the object name is valid, the object full path is displayed.
5. Click **Add**.
6. Click **Save** or **Save and Close**.

Configuring Screen Pop Pages: Explained

You can configure screen pop pages to display pages of information that can aid an agent to start a customer interaction efficiently. For example, you can configure a screen pop page to display information about an open ticket logged by the call-in customer. Configuring the screen pop page has the advantage of expediting the call, while not having to ask the customer for basic information.

You can create screen pop pages for ready-to-use standard business objects or for user-defined objects. To create screen pop for user-defined objects, you must first create the objects, define the fields, pages, and other elements using Application Composer. For more information about creating user-defined business objects using Application Composer, see Oracle Sales Cloud Extending Sales guide.

Configuring a screen pop page includes the following steps to be performed in Setup and Maintenance.

1. Configure business objects. These business objects are associated with standard or user-defined objects. Standard objects include ready-to-use objects, such as Service Requests or Queues, and the user-defined objects are created by the user.
2. Create tokens. Tokens are associated with the attributes of a business object. For example, you can create a token called **SVC_INVOICE_DATE** and associate it with the **Invoice_Date** field of the Invoices business object.
3. Map pages. Mapping associates a screen pop page with the pages of the underlying standard or user-defined object.

4. Create rules. Rules determine the page that is displayed when passing a token. Rules are defined in order of priority. For example, create rules that, if a service request number is available, display the service request page. If no service request number is available, but there is a contact identified, display the Edit Contact page. If no service request or contact information is available, display the Create Contact page.

The following figure illustrates the process of configuring screen pop pages:

1. Configure business objects
2. Create tokens
3. Map pages
4. Create rules
Configuring Screen Pop Pages: Procedure

This topic describes how to configure screen pop pages. Configuring a screen pop page includes the following steps:

1. Creating a token
2. Mapping a page
3. Defining rules

To configure a screen pop page:

1. Sign in to Oracle Cloud applications as a Service Administrator.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Communication Channels functional area, and then select Manage Screen Pop Configuration task.

The Screen Pop Configuration page is displayed.

Creating a Token

To create a token:

1. Click the Tokens tab.
   Tokens tab lists standard and user-defined tokens. The standard or the predefined tokens cannot be modified or deleted.
2. Click the User-Defined Tokens tab.
3. To add a token, click the add icon.
   A token consists of the following details:
   - Name: Any alphanumeric value that represents the token
   - Token Code: Unique code that is used to represent a token
   - Description: String to provide extra information about the token
   - Object Name: Business object to which the token is associated. Object Name is an optional value.
   - Object Attribute: Attribute of a business object that is associated with the token. Attribute is an optional value.
4. Specify the values for the token and click Save.

Mapping Screen Pop Page

After creating the token, map a screen pop page to an existing page of the associated business object. To map a screen pop page:

1. On the Screen Pop Page Configuration page, click the Pages tab.
2. To create a page mapping, click the add icon.
3. Select a Standard or a User-Defined type.
4. Select an Object Name.
5. Enter a Page Name.
6. Enter a **Page Title** for the screen pop page. A title can contain a title prefix, the name of the token, and a title suffix. One of these values is mandatory.

7. If you have selected a User-Defined type of mapping, the Page Path is displayed automatically when you select the page. However, for a standard mapping, you must specify the page path. After specifying a page path, click **Inspect** to validate the page path and to list the page parameters.

8. In the Page Parameters Mapping section, associate required parameter with a token or a user-defined value. Based on the input that is passed to one or more page parameters, the information is displayed in the page. For example, based on the invoice number parameter, the invoice details screen is displayed.

9. Click **Save**.

The Show Only Used check box lists only those pages that are specified in a rule.

### Defining Screen Pop Rules

Screen pop configuration rules determine which set of rules must be applied when a screen pop logic is invoked. Different screen pop rules can be invoked based on a number of different variables, such as application classification and channel. Based on the input parameters, you can choose from a number of different pages to screen pop to the agent. For example, pages such as the Contact Edit, Account Edit, and Service Request edit pages can be displayed to the customer. Additionally, you can choose to create an object, such as a service request. This framework also allows for user-defined objects to be presented to the agent as part of the screen pop process.

Define rules in order of priority to display a screen pop page, when an associated token value is available. If a rule in a higher priority is not satisfied, the next in the order is checked. To define rules:

1. On the **Screen Pop Page Configuration** page, click the **Rules** tab.
2. First, create a rule set. A rule set consists of one or more rules that are defined in an order of priority. To create a rule set:
   a. Click the add icon. You can also select **Duplicate** from the **Actions** menu to duplicate an existing rule set.
   b. Enter a **Rule Set Name**.
   c. Select an **Application Classification** to which the rule set belongs.
   d. A rule set is **Active** by default. To deactivate a rule set, clear the **Active** option.
   e. Enter a description for the rule set.
3. Next, add rules to the rule set. To add rules:
   a. Click the add icon.
   b. The priority column displays the order of priority in which the rules are checked. You can change the priority by clicking up arrow and down arrow icons.
   c. A rule is enabled by default. Clear the **Enable** option to disable a rule.
   d. Select the communication channel to which the rule is applied. For example, a rule is applied only when an agent receives a service phone call or when there is a chat alert from a customer. You can add or modify the channels list by modifying the associated lookup values.
4. Select a **Token** name.
5. Select a page to display, when a token value is available.
6. Click **Save**.
7. When you’re finished creating the rule sets, click **Done**.
Creating Lookup Filters: Procedure

When a customer makes a service call, customer provides information, such as Customer ID, Date of Birth, and First Name through an Interactive Voice Response (IVR) tool. Lookup filters are used to identify and verify customers based on the information given. This information is passed in as tokens to extract a customer record from the database.

Lookup sets are grouped by application classification. They contain one or more lookup filters defined in an order of priority, and each lookup filter is associated with a token. Customer details are extracted from the database based on the availability of the token value, starting from the token of highest priority.

To create a new lookup filter, you can duplicate an existing one and make modifications. To create a new lookup filter from the scratch, perform the following steps:

1. Sign in to the Oracle Cloud application as a Service Administrator.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Communication Channels functional area, and then select Manage Lookup Filters task.
5. To create a lookup set, click the add icon.
   a. Enter a name.
   b. Select an Application Classification from drop-down list menu.
   c. Select the Active option to activate a lookup set.
   d. Enter a Description.
6. Add filters within the newly created lookup set. To add filters, click the add icon.
   a. Enter a name.
   b. Select the interaction channel from the Channels drop-down list. The filter is applied to the selected interaction channel.
   c. Select a Token from the drop-down list. When you select a token, the associated object name is displayed in the Object Name column.
   d. Add more filters if required.

You can reorder the filters by clicking the up and down arrow icons. Alternatively, click the Actions drop-down list and select Move Up or Move Down option.

7. Click Save or Save and Close.

Configuring Call Flow Parameters: Procedure

The call flow parameters determine the default and administrator-defined actions to be performed for a customer interaction. Customer interactions include calls or chats with the customer. Administrator-defined settings override the default settings. You can specify the settings listed in the following table:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Required</td>
<td>Yes/No</td>
<td>Specifies if contact details of a customer must be displayed or not</td>
</tr>
<tr>
<td>Setting</td>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Contact Verification</td>
<td>Yes/No</td>
<td>Specifies whether the customer details must be verified at the beginning of the call or chat</td>
</tr>
<tr>
<td>Screen Pop</td>
<td>Yes/No</td>
<td>Specifies if a screen pop page must be displayed or not</td>
</tr>
<tr>
<td>Wrap Up</td>
<td>Yes/No/Server Driven</td>
<td>Specifies if a customer interaction requires wrap up or not. Select the Server Driven option to get notified by the toolbar if an interaction must be wrapped up or not.</td>
</tr>
</tbody>
</table>

To configure call flow parameters:

1. Sign in to Oracle Cloud application as a Service Administrator.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Communication Channels functional area, and then select Configure Call Flow Parameters task.
5. Specify default settings for all interaction channels in the Default Settings section.
6. In the Administrator-Defined Settings section, specify the settings for the individual interaction channels. These settings override the default settings. For example, if the default setting is to not verify a contact, but for a service phone channel the Contact Verification is set to Yes, the contact of the caller is verified.

Note: If you specify Yes in the Wrap Up setting for the Chat channel, the wrap up details and the transcript for every chat are saved.

7. Click Save or Save and Close.

Modifying the Default Contact Search: Procedure

You can modify the default contact search page by adding or removing fields. To modify the default contact search page, perform the following procedure.

1. Activate a sandbox.
   
   For more information about activating a sandbox, refer to the Using Sandboxes link in the Related Topics.  

2. In the Setting and Actions menu, select Edit Pages.  

3. In the Edit Pages window, select Site and click OK.  

   The Page Composer is displayed.  

4. Select the Add Content tab.  

5. Select the Contact Search option from the Tools group in the Navigator.  

6. In the Contact Search page, click Advanced.  

7. Add custom fields and click Save.  

8. Specify a search name and select Set as Default option to set the search as default.  

9. Click OK and select the Contact Search window.  

10. Close the Contact Search window.  

11. In the Editing User Interface screen, open the Select tab.
12. Click **Service** and then click **Edit Component**.
13. In the **Edit Components** window, click **OK**.
14. Verify the changes and publish the sandbox.

**Related Topics**

- Searching for a Contact: Procedure
- Using Sandboxes: Explained
9 Understanding CTI Media Toolbar APIs

Media Toolbar API: Overview

Media toolbar APIs are a set of JavaScript functions that are delivered by Oracle to enable Computer Telephony Integration (CTI) partners to integrate their media toolbars with Oracle Engagement Cloud. These APIs facilitate communication between the media toolbar and Oracle Engagement Cloud to exchange the following in response to inbound or outbound interaction requests:

- Configuration information
- Events
- Payload data

When enabled, Oracle Engagement Cloud loads the toolbar and the media toolbar APIs in an iFrame within the browser.

How to Load the Media Toolbar API

Toolbar API modules are available under the `svcMca.tlb.api` namespace to enable quick and clear identification, and to avoid conflicts with similar JavaScript functions. To load the API JavaScript file, concatenate the following two parameters to the iFrame source toolbar URL:

- `oraParentFrame`: Represents the origin of Oracle Engagement Cloud, including the protocol and the port. For example, `https://mypodinstance.oraclecloud.com:12345`.
- `oraApiSource`: Represents the path from where the toolbar API JavaScript file can be loaded, including the leading slash. For example, `/service/js/mcaInteractionV1.js`.

Media Toolbar API: Methods

Media toolbar APIs are divided into the following methods:

- Configuration methods: These methods enable the toolbar and the Oracle Engagement Cloud module to exchange configuration information regarding the features that are available in toolbar implementation and what features are supported by Oracle Engagement Cloud. These methods allow for configuration or changes to be saved in Oracle Engagement Cloud and to be retrieved by the toolbar for usage.
- Interaction methods: These methods provide notification on the events occurring on the toolbar. These methods also exchange information that is used by the toolbar or Oracle Engagement Cloud.
- Toolbar windows management methods: These methods manage the toolbar windows and facilitate communication between the windows.
- Event listeners: These methods allow the toolbar to register listeners for events of interest, that are triggered by Oracle Engagement Cloud.
The APIs are designed to provide a non-blocking interaction between the toolbar and Oracle Engagement Cloud. Each of the methods contains a callback function parameter that is invoked when a method on Oracle Engagement Cloud is complete. The callback function parameter notifies the toolbar of the outcome.

Media Toolbar API: Introduction

Media toolbar APIs are a set of JavaScript functions that are delivered by Oracle to enable Computer Telephony Integration (CTI) partners to integrate their media toolbars with Oracle Engagement Cloud. These APIs facilitate communication between the media toolbar and Oracle Engagement Cloud to exchange the following in response to inbound or outbound interaction requests:

- Configuration information
- Events
- Payload data

When enabled, the toolbar and the media toolbar APIs are loaded in an iFrame within the browser.

Scope

The media toolbar API is to be used with all toolbars that handle communication between agents and customers. Toolbar loads the APIs that are implemented in a JavaScript file. The APIs are designed to

- Communicate with the Oracle Engagement Cloud ADF page
- Exchange events that contain information to be used by the toolbar to present relevant data to the agents
- Be used by the Oracle Engagement Cloud to execute data retrieval, screen pop, and interaction update

Although media toolbar APIs are generic to all media channels, the 11.13.0.0 release supports only Phone and integration with CTI partner systems.

API Classification

The media toolbar APIs facilitate the initial configuration of the loaded toolbar, and enable you to setup the functionality that are supported by the partner toolbar and Oracle Engagement Cloud. The design targets the following three main flows of the toolbar interaction:

- Initial configuration of the partner toolbar and communication with Oracle Engagement Cloud
- Inbound communication handling
- Outbound communication handling

APIs are independent of communication channels, but uses the channel information as a parameter that is passed to Oracle Engagement Cloud, to help differentiate when a toolbar supports multiple channels. The channels are configured using the Functional Setup Manager where channels can be defined. The APIs support both preconfigured and user-defined channels.

Data Flow Methods

The following figures illustrate the flow of the methods that are to be called by the toolbar. The method calls can be mandatory or optional. If a call is mandatory, the toolbar implementation has to make the call once for each communication
event, unless specified otherwise. If a call is optional, the toolbar implementation can make a call whenever necessary, depending on the action needs.

The following figure shows the legend

![Legend Diagram]

The following figure illustrates the configuration workflow.

![Configuration Workflow Diagram]

The following figure shows the inbound call flow.
The following figure shows the outbound call flow.
The following figure shows the end of call workflow.
Configuration APIs: Explained

The configuration APIs that enable the toolbar and Oracle Engagement Cloud to exchange configuration data are specified in the following table.
Method | Usage
--- | ---
getConfiguration | Method to set up the toolbar. Provides a list of features that are supported by Oracle Engagement Cloud.
disableFeature | Method to disable a feature that the toolbar does not support, so that the application does not use the feature.
readyForOperation | Method to notify Oracle Engagement Cloud if the toolbar is ready for use. This also means that the agent is logged on to the toolbar and ready to handle the communication.

getConfiguration ([configType], callback)

Call this mandatory method to obtain configuration information. The configuration information enables the toolbar to evaluate the features that are supported by Oracle Engagement Cloud. Based on the evaluation, the toolbar can inform Oracle Engagement Cloud of the features that must be disabled they’re not implemented by the toolbar. For more information, see the disableFeature API method. The configuration information also contains details about the initial toolbar dimensions, status, tokens, pages, and the lookup objects for the rule sets used in configuration.

The getConfiguration method details are as follows

- Parameters:
  - `callback`: Method callback to let the caller toolbar know of the results, and get the configuration information which is the JSON formatted string.
  - `configType`: Type of configuration information to be returned by Oracle Engagement Cloud. Possible values are TOOLBAR, TOKENS, PAGES, LOOKUP_OBJECTS, ALL, or FA_TOKEN. This parameter is optional and has a default value of TOOLBAR.

- Return value: Agent name and the following information based on the configType parameter that is passed
  - `TOOLBAR`: Returns Engagement Cloud supported features and are provided in the response to let the toolbar know what functionality can be used. These are preconfigured values. This parameter also returns Toolbar information, such as if a toolbar is enabled, and the toolbar dimensions. This information is retrieved from configuration done using the Functional Setup Manager. These values can be modified.
  - `TOKENS`: Returns information about both system and user-defined tokens, with code, name, related entity, attribute of the related entity (if configured). The information is retrieved using the Functional Setup Manager. These values can be modified.
  - `PAGES`: Returns information about both system and user-defined screen pop pages, with code, name, parameters, and associated tokens. The information is retrieved using the Functional Setup Manager. These values can be modified.
  - `LOOKUP_OBJECTS`: Returns information about business objects that are used to perform reverse lookup. The information is retrieved using the Functional Setup Manager. These values can be modified.
  - `ALL`: Returns all of information as the other configType values, except that of FA_TOKEN type.
  - `FA_TOKEN`: Returns a JWT token to be used by toolbar implementation to authenticate against Engagement Cloud when calling the REST APIs. A specific call to this configType is required to obtain the JWT token.

Sample code to call the getConfiguration method

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
```
disableFeature (features, callback)

This method informs Oracle Engagement Cloud that some of the functionality that is available must be disabled, because the toolbar has not implemented the functionality. By default, the features sent by the getConfiguration API method are considered enabled. Method call is optional and can be called multiple times. The method details are as follows:

- **Parameters**
  - **features**: String that contains the comma-separated feature names. For a list of features, see API Features: Explained.
  - **callback**: Method callback to let the caller toolbar know of the results

- **Returns**: None

Sample code to call the method

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>
<body>
<script type="text/javascript">
function disableOutgoingCall() {
  svcMca.tlb.api.disableFeature('OUTBOUND_CALL', function (response) {
    if (response.result == 'success') {
      alert('Success! Feature disabled!');
    } else {
      alert('Operation finished with error: ' + response.error);
    }
  });
}</script>
<input type="button" value="Configuration" onclick="callGetConfiguration()"/>
</body>
</html>
```
readyForOperation (readiness, callback)

This method notifies Oracle Engagement Cloud that the toolbar is ready for operation. A minimum of one call to this method is mandatory to notify the readiness of the toolbar. The toolbar is NOT READY by default, until this method is called with readiness parameter set to true. The method details are as follows.

- **Parameters**
  - readiness: Boolean value to specify if the toolbar is ready or not to operate
  - callback: Method callback to let the toolbar know that Oracle Engagement Cloud received the readiness status

- **Returns**: None

Sample code to call the method

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
<script type="text/javascript">
function setReady() {
  svcMca.tlb.api.readyForOperation(true, function (response) {
    if (response.result == 'success') {
      alert('Success!');
    } else {
      alert('Operation finished with error: ' + response.error);
    }
  });
}
</script>
</head>
<body>
<input type="button" value="Get state" onclick="getDisplayState()"/>
</body>
</html>
```

### Interaction API: Explained

Interaction API facilitates communication between the toolbar and Oracle Engagement Cloud during an interaction. The API triggers customer identification by using reverse lookup, screen pop and user-defined tokens. All methods of the interaction API have channel for which they make the call as a parameter. The channels are configured using the Functional Setup Manager. For a list of out-of-the-box channel names, see Channel Type Data: Explained. The following table lists the Interaction API methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>newCommEvent</td>
<td>Notifies Oracle Engagement Cloud that the toolbar received a new CommEvent. This means that new communication has been assigned to or was initiated by the current agent. Call is mandatory.</td>
</tr>
<tr>
<td>Method</td>
<td>Usage</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>startCommEvent</td>
<td>Notifies Oracle Engagement Cloud that the toolbar has started a CommEvent, where an agent has accepted the event.</td>
</tr>
<tr>
<td>transferCommEvent</td>
<td>Notifies Oracle Engagement Cloud that the toolbar is transferring the CommEvent to another agent.</td>
</tr>
<tr>
<td>conferenceCommEvent</td>
<td>Notifies Oracle Engagement Cloud that the toolbar is starting a conference based on the current CommEvent.</td>
</tr>
<tr>
<td>closeCommEvent</td>
<td>Notifies Oracle Engagement Cloud that the toolbar has closed the CommEvent. This means that the agent has rejected or ended the event. Call is mandatory.</td>
</tr>
<tr>
<td>setCommEventTranscript</td>
<td>Transfers the transcript for the event, such as chat, SMS, or email, to the application. The method can be included in the wrap up or can be added as a separate item.</td>
</tr>
<tr>
<td>invokeScreenPop</td>
<td>Sends request to Oracle Engagement Cloud to do a specific screen pop irrespective of the configured rules.</td>
</tr>
<tr>
<td>getCustomerData</td>
<td>Retrieves customer or related data.</td>
</tr>
</tbody>
</table>

**newCommEvent (channel, appClassification, eventId, inData, [lookupObject], callback, channelType)**

Toolbar calls this method to inform Oracle Engagement Cloud that a new communication event was received. This is the first API call that the toolbar has to make for a communication event. This method needs to provide a unique identifier or the media event identifier to be used for a communication event through the `eventId` parameter. The primary purpose of the call is to identify customers by using reverse lookup functionality. The toolbar passes all the information from IVR system, by using tokens. All customer information retrieved is sent back to the toolbar using system tokens. Method call is mandatory.

Method details are as follows.

- **Parameters**
  - `channel`: Name of the channel for which the method is called.
  - `appClassification`: Name of the application classification defined in the system for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of out-of-the-box values, see Application Classification Code: Explained.
  - `eventId`: Vendor generated media event identifier. The same parameter value must be used for all method invocations for a communication. This parameter is stored with internal interaction record.
  - `inData`: Object containing information about incoming events for customer identifiable data. The `inData` object has attributes with the names of system or user-defined defined tokens
  - `callback`: Method callback to let the caller toolbar know of the results.
  - `lookupObject`: Business object that is used to run the reverse lookup. This parameter is optional and a default value of Contact is used for customer information. For more information about the list of system lookup objects, see System Business Objects: Explained.
  - `channelType`: The type of channel for which the method is being called. If the channel is PHONE and the channelType is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.
The response object has following member attributes

- **result**: Represents the status of the running of the method on the server side. This attribute contains success or error.
- **outData**: Represents an object containing possible identified data from Oracle Engagement Cloud for the contact and for the organization. This attribute contains the names of system tokens.
- **error**: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

**Returns**: None

**Note**: The `newCommEvent` method triggers the creation of an interaction object that contains information related to communication, and the objects that are related to communication. The identifier for the interaction is passed back with the callback response as the value for token `SVCMCA_INTERACTION_ID`. Subsequent method invocations must send this token and the value, along with other IVR data to Oracle Engagement Cloud.

When `newCommEvent` method is invoked for a transfer, the toolbar sends the IVR data and the originating communication interaction ID using the `SVCMCA_PARENT_INTERACTION_ID` token to Oracle Engagement Cloud, to associate the communication events. In addition, the `SVCMCA_2NDARY_CALL_TYPE` token must also be specified to indicate whether this call is a Transfer, a Consultation, or a Conference.

Sample code to call the method.

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>
<body>
<input type="button" value="Get customer data" onclick="inboundCallNotification()"/>
</body>
</html>
```

`startCommEvent (channel, appClassification, eventId, inData, callback, channelType)`

Toolbar must call this method when the agent accepts a new communication that was triggered by the previous `newCommEvent()` call. This method invokes the Identify Contact flow, then triggers the Customer Validation flow, and then perform the screen pop. This method call is mandatory when agent accepts a new communication.

- The Identify Contact flow queries the communication data to determine if a contact has already been identified. If not, the Contact Search window is displayed allowing the agent to search for or create a contact. Once found, the contact information is returned to the toolbar. Toolbars can bypass this flow by identifying the contact prior...
to the `startCommEvent` call or by passing the `SVCMCA_BYPASS_IDENTIFY_CONTACT`. Toolbars can then use the `getCustomerData()` method to manually request contact identification.

- Customer Verification is an optional flow that opens the customer verification page. If the agent does not verify the caller, no screen pop is performed. Customer Verification can be bypassed by passing in the `SVCMCA_BYPASS_CUSTOMER_VERIFICATION` token in the call data.

- Screen pop is run using the rules setup in the system. If the toolbar would like to prevent the screen pop, they can add the `SVCMCA_BYPASS_AUTO_SCREENPOP` token into the call data, and use the `invokeScreenPop()` method to manually trigger the pop.

Callback passed with `startCommEvent` can be invoked multiple times if Identify Contact and Customer Verification flows are involved. The `outData` information from response may contain updated information about a customer. Method details are as follows.

- **Parameters**
  - `channel`: Name of the channel for which the method is called.
  - `appClassification`: Name of the application classification defined in the system for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of out-of-the-box values, see Application Classification Code: Explained.
  - `eventId`: Vendor generated media event identifier. The same parameter value must be used for all method invocations for a communication. This parameter is stored with internal interaction record.
  - `inData`: Object containing information about incoming events for customer identifiable data. The `inData` object has attributes with the names of system or user-defined tokens.
  - `callback`: Method callback to let the caller toolbar know of the results.
  - `lookupObject`: Business object that is used to run the reverse lookup. This parameter is optional and a default value of Contact is used for customer information. For more information about the list of system lookup objects, see System Business Objects: Explained.
  - `channelType`: The type of channel for which the method is being called. If the channel is PHONE and the `channelType` is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.

- **The response object has following member attributes**
  - `result`: Represents the status of the running of the method on the server side. This attribute contains success or error.
  - `outData`: Represents an object containing possible identified data from Oracle Engagement Cloud for the contact and for the organization. This attribute contains the names of system tokens.
  - `error`: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

- **Returns**: None

Sample code to call the method.

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>

<script type="text/javascript">
function startCommEvent() {
    var inData = {};
    inData.SVCMCA_ANI = '5551234';
    inData.SVCMCA_CONTACT_ID = '1234567890';
    inData.SVCMCA_CONTACT_NAME = 'John Smith';
    inData.createSr = 'yes';
    svcMca.tlb.api.startCommEvent('PHONE', 'ORA_SERVICE', '12345-1234-67890', inData, function (response) {
        // Call back function
    });
}
</script>
</html>
```
if (response.result == 'success') {
    console.log('Created service request: ' + response.SVCMCA_SR_NUMBER);
    alert('Created service request: ' + response.SVCMCA_SR_NUMBER);
} else {
    alert('Operation finished with error: ' + response.error);
}
,'ORA_SVC_PHONE');
</script>
</head>
<body>
<input type="button" value="Call started" onclick="startCommEvent()"/>
</body>
</html>

closeCommEvent (channel, appClassification, eventId, inData, reason, callback, channelType)

Toolbar calls this method when the communication is rejected, disconnected, ends normally, or enters the wrap-up mode. Call to this method is mandatory when startCommEvent is called when communication is accepted. This method is called with following the reason values.

- **REJECT** when an agent rejects the communication
- **WRAPUP** to indicate that Oracle Engagement Cloud must wrap up the call by displaying the Wrap Up window
- **ENDCOMMUNICATION** to indicate that the communication ended and if Oracle Engagement Cloud is in wrap-up mode it needs to close the wrap-up UI page

Method details are as follows.

- **Parameters**
  - **channel**: Name of the channel for which the method is called.
  - **appClassification**: Name of the application classification defined in the system for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of out-of-the-box values, see Application Classification Code; Explained.
  - **eventId**: Vendor generated media event identifier. The same parameter value must be used for all method invocations for a communication. This parameter is stored with internal interaction record.
  - **inData**: Object containing information about incoming events for customer identifiable data. The inData object has attributes with the names of system or custom defined tokens.
  - **callback**: Method callback to let the caller toolbar know of the results.
  - **lookupObject**: Business object that is used to run the reverse lookup. This parameter is optional and a default value of Contact is used for customer information. For more information about the list of system lookup objects, see System Business Objects: Explained.
  - **channelType**: The type of channel for which the method is being called. If the channel is PHONE and the channelType is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.

- **The response object has following member attributes**
  - **result**: Represents the status of the running of the method on the server side. This attribute contains success or error.
  - **outData**: Represents an object containing possible identified data from Oracle Engagement Cloud for the contact and for the organization. This attribute contains the names of system tokens.
  - **error**: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error messages.

- **Returns**: None
Sample code to call the method.

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
<script type="text/javascript">
function closeCommEvent() {
  var inData = {};
  inData.SVCMCA_ANI = '5551234';
  inData.SVCMCA_CONTACT_ID = '1234567890';
  inData.SVCMCA_CONTACT_NAME = 'John Smith';
  svcMca.tlb.api.closeCommEvent('PHONE', 'ORA_SERVICE', '12345-1234-67890', inData, 'WRAPUP', function (response) {
    if (response.result == 'success') {
      alert('Success! Call ended.');
    } else {
      alert('Operation finished with error: ' + response.error);
    }
  }, 'ORA_SVC_PHONE');
}
</script>
</head>
<body>
<input type="button" value="End call" onclick="closeCommEvent()"/>
</body>
</html>

transferCommEvent and conferenceCommEvent

• `transferCommEvent (channel, appClassification, eventId, inData, callback, channelType)`: Toolbar calls this method to inform Oracle Engagement Cloud that a transfer of the communication has been initiated. Oracle Engagement Cloud records the notification, but does not respond with any action. After a call to this method, toolbar calls `closeCommEvent()` method with the reason `TRANSFERRED`.

• `conferenceCommEvent (channel, appClassification, eventId, inData, callback, channelType)`: Toolbar calls this method to inform Oracle Engagement Cloud that a conference has been initiated. Oracle Engagement Cloud records the notification without any action.

Details of the `transferCommEvent` and `conferenceCommEvent` methods are as follows.

• Parameters

  o `channel`: Name of the channel for which the method is called.
  o `appClassification`: Name of the application classification defined in the system for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of out-of-the-box values, see Application Classification Code: Explained.
  o `eventId`: Vendor generated media event identifier. The same parameter value must be used for all method invocations for a communication. This parameter is stored with internal interaction record.
  o `inData`: Object containing information retrieved from IVR or other systems that process incoming events for customer identifiable data and data received from previous calls to toolbar API method to be used by Oracle Engagement Cloud to process the event. The `inData` object has attributes with the names of system or custom defined tokens.
  o `callback`: Method callback to let the caller toolbar know of the results.
  o `channelType`: The type of channel for which the method is being called. If the channel is PHONE and the `channelType` is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.
The response object has following member attributes:

- **result**: Represents the status of the running of the method on the server side. This attribute contains success or error.
- **error**: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error messages.

Returns: None

Sample code to call the `transferCommEvent` method.

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
<script type="text/javascript">
function transferCommEvent() {
    var inData = {};
    inData.SVCMCA_ANI = '5551234';
    inData.SVCMCA_CONTACT_ID = '1234567890';
    inData.SVCMCA_CONTACT_NAME = 'John Smith';
    inData.transferedId = '5550987';
    svcMca.tlb.api.transferCommEvent('PHONE', 'ORA_SERVICE', '12345-1234-67890', inData, function (response) {
        if (response.result == 'success') {
            alert('Success! Call transferred to other agent.');
        } else {
            alert('Operation finished with error: ' + response.error);
        }
    },'ORA_SVC_PHONE');
</script>
</head>
<body>
<input type="button" value="Notify transfer" onclick="transferCommEvent()"/>
</body>
</html>
```

Sample code to call the `conferenceCommEvent` method.

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
<script type="text/javascript">
function conferenceCommEvent() {
    var inData = {};
    inData.SVCMCA_ANI = '5551234';
    inData.SVCMCA_CONTACT_ID = '1234567890';
    inData.SVCMCA_CONTACT_NAME = 'John Smith';
    inData.conferenceParticipantIds = ['5550987', '5552468'];
    svcMca.tlb.api.conferenceCommEvent('PHONE','ORA_SERVICE','12345-1234-67890',inData, function (response){
        if (response.result == 'success') {
            alert('Success! Call conference started.');
        } else {
            alert('Operation finished with error: ' + response.error);
        }
    },'ORA_SVC_PHONE');
</script>
</head>
<body>
<input type="button" value="Notify conference started" onclick="conferenceCommEvent()"/>
</body>
</html>
```
invokeScreenPop (channel, appClassification, eventId, pageCode, pageData, callback, channelType)

Toolbar can invoke this method to trigger the display of a screen pop page. The toolbar provides the page code that is derived from the configuration, and the required tokens configured as parameters for the page. Method details are as follows.

- **Parameters**
  - **channel**: Name of the channel for which the method is called.
  - **appClassification**: Name of the application classification defined in the system for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of out-of-the-box values, see Application Classification Code: Explained.
  - **eventId**: Vendor generated media event identifier. The same parameter value must be used for all method invocations for a communication. This parameter is stored with internal interaction record.
  - **pageCode**: Page code defined in the system to invoke the screen pop. For more information, see System Pages: Explained.
  - **pageData**: Object containing information retrieved from IVR or other systems that process incoming events for customer identifiable data and data received from previous calls to toolbar API method to be used by Oracle Engagement Cloud to process the event. The pageData object has attributes with the names of system or user-defined tokens. When the method is called, the transcript value is stored in the SVCMCA_TRANSCRIPT token.
  - **callback**: Method callback to let the caller toolbar know of the results.
  - **channelType**: The type of channel for which the method is being called. If the channel is PHONE and the channelType is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.

- **The response object has following member attributes**
  - **result**: Represents the status of the running of the method on the server side. This attribute contains success or error.
  - **outData**: Represents an object containing the system or user-defined tokens returned by action.
  - **error**: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

- **Returns**: None

Sample code to call the method.

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>
<script type="text/javascript">
function invokeScreenPop() {
    var inData = {};
    inData.SVCMCA_ANI = '5551234';
    inData.SVCMCA_CONTACT_ID = '1234567890';
    inData.SVCMCA_CONTACT_NAME = 'John Smith';
    svcMca.tlb.api.invokeScreenPop('PHONE', 'ORA_SERVICE', '12345-1234-67890', 'Edit_Contact', inData, function (response) {
        if (response.result == 'success') {
            alert('Success! Edit Contact page popped.');
        } else {
            alert('Operation finished with error: ' + response.error);
        }
    }, 'ORA_SVC_PHONE');
}
</script>
</html>
```
getCustomerData (channel, appClassification, eventId, inData, [lookupObject], callback, channelType)

Toolbar calls this method to perform a reverse lookup during a communication lifecycle. This method call is optional. Method details are as follows.

- **Parameters**
  - `channel`: Name of the channel for which the method is called.
  - `appClassification`: Name of the application classification defined in the system for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of out-of-the-box values, see Application Classification Code: Explained.
  - `eventId`: Vendor generated media event identifier. The same parameter value must be used for all method invocations for a communication. This parameter is stored with internal interaction record.
  - `inData`: Object containing information about incoming events for customer identifiable data. The `inData` object has attributes with the names of system or user-defined tokens.
  - `callback`: Method callback to let the caller toolbar know of the results.
  - `lookupObject`: Business object that is used to run the reverse lookup. This parameter is optional and a default value of Contact is used for customer information. For more information about the list of system lookup objects, see System Business Objects: Explained.
  - `channelType`: The type of channel for which the method is being called. If the channel is PHONE and the `channelType` is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.

- **The response object has following member attributes**
  - `result`: Represents the status of the running of the method on the server side. This attribute contains success or error.
  - `outData`: Represents an object containing possible identified data from Oracle Engagement Cloud for the contact and for the organization. This attribute contains the names of system tokens.
  - `error`: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error messages.

- **Returns**: None

Sample code to call the method.

```javascript
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>
<body>

function getCustomerData() {
  var inData = {};
  inData.SVCMCA_ANI = '5551234';
  inData.SVCMCA_CONTACT_ID = '1234567890';
  svcMca.tlb.api.getCustomerData('PHONE', 'ORA_SERVICE', '12345-1234-67890', inData, function (response) {
    if (response.result == 'success') {
      console.log('Customer: ' + response.outData.SVCMCA_CONTACT_NAME + ' (' + response.outData.SVCMCA_CONTACT_ID + ')');
      console.log('Account:' + response.outData.SVCMCA_ORG_NAME + '(' + response.outData.SVCMCA_ORG_ID + ')');
      alert('Success! Results available in log.');
    } else {
      alert('Operation finished with error: ' + response.error);
    }
  });
}
</body>
</html>
```
Window Management API: Explained

Window management API is designed to facilitate additional toolbar management from the main horizontal toolbar. Oracle Engagement Cloud acts as a message bus for the communication between horizontal, vertical, and notification toolbars, and enables opening and closing of vertical toolbar.

The following table lists the Window management APIs and their usage.

<table>
<thead>
<tr>
<th>Method</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>openFloatingToolbar</td>
<td>Requests Oracle Engagement Cloud to programmatically open a floating toolbar.</td>
</tr>
<tr>
<td>isFloatingToolbarOpen</td>
<td>Requests Oracle Engagement Cloud about floating bar being open already or not.</td>
</tr>
<tr>
<td>closeFloatingToolbar</td>
<td>Requests Oracle Engagement Cloud to programmatically close floating toolbar.</td>
</tr>
<tr>
<td>postToolbarMessage</td>
<td>Sends a message from one toolbar to the other by using Oracle Engagement Cloud as a message bus. For example, sending a message from the horizontal toolbar to the floating toolbar.</td>
</tr>
<tr>
<td>onToolbarMessage</td>
<td>Registers a callback for the messages exchanged between toolbars.</td>
</tr>
<tr>
<td>setToolbarDimensions</td>
<td>Sets toolbar dimensions for horizontal and vertical toolbars.</td>
</tr>
</tbody>
</table>

`openFloatingToolbar (url, height, width, inData, callback)`

Toolbar calls this method to programmatically open an additional toolbar. This method call is optional.

Method details are as follows.

- **Parameters**
  - **url**: URL of the page to be opened in the iFrame of the floating toolbar. If this value is null, Oracle Engagement Cloud tries to find the URL in the configuration.
  - **height**: Initial height of the additional toolbar
  - **width**: Initial width of the additional toolbar
  - **inData**: Object containing information about incoming events for the floating dialog box. The `inData` object has attributes with the names of system or user-defined tokens.
  - **callback**: Method callback to let the caller toolbar know of the results.
• The response object has following member attributes
  ◦ **result**: Represents the status of the running of the method on the server side. This attribute contains success or error.
  ◦ **error**: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

• Returns: None

⚠ **Note**: The width and height attribute values are limited to 470 pixels. Using a higher value in method attributes returns an error message.

**closeFloatingToolbar (callback)**

Toolbar calls this method to close an additional toolbar. This method call is optional.

Method details are as follows.

• Parameters
  ◦ **callback**: Method callback to let the caller toolbar know of the results.

• The response object has following member attributes
  ◦ **result**: Represents the status of the running of the method on the server side. This attribute contains success or error.
  ◦ **error**: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

• Returns: None

**postToolbarMessage (messagePayload, callback)**

Toolbar calls this method to send messages from one toolbar to another. Oracle Engagement Cloud sends payload message to all opened toolbars, except for the toolbar that sent the message.

Method details are as follows.

• Parameters
  ◦ **messagePayload**: Payload that has to be passed to other toolbars.
  ◦ **callback**: Method callback to let the caller toolbar know of the results.

• The response object has following member attributes
  ◦ **result**: Represents the status of the running of the method on the server side. This attribute contains success or error.
  ◦ **error**: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

• Returns: None
onToolbarMessage (callback)
This method registers a callback for the messages that are exchanged between toolbars. The callback is registered only for the toolbar that sent the message.

Method details are as follows.

- Parameters
  - callback: Method to be called when a message is received from other toolbars.

- The response object has following member attributes
  - result: Represents the status of the running of the method on the server side. This attribute contains success or error.
  - messagePayload: Payload from a different toolbar.
  - error: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

- Returns: None

setToolbarDimensions (barType, height, width, callback)
Toolbar calls this method to change the current dimensions of the toolbar represented by barType parameter. If the wanted screen space is not available to change the dimension, the maximum screen space that is available is allocated and an error message is sent back to toolbar. This method call is optional.

Method details are as follows.

- Parameters
  - barType: Type of toolbar to apply the changes. The values can be MAIN_HORIZONTAL, FLOAT_VERTICAL, or NOTIFICATION.
  - height: Number that represent wanted toolbar height.
  - width: Number that represent wanted toolbar width. This parameter does not apply to toolbars of type MAIN_HORIZONTAL.
  - callback: Method callback to let the caller toolbar know of the results.

- The response object has following member attributes
  - result: Represents the status of the running of the method on the server side. This attribute contains success or error.
  - error: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error Messages: Explained.

- Returns: None

> Note: The width and height attribute values are limited depending on the toolbar for which the method is invoked. For horizontal toolbars, the height has a maximum value of 70 pixels, and width value is ignored. For vertical toolbars, the height and width must not be more than 470 pixels. For notification toolbars, the height has a maximum value of 180 pixels and width has a maximum value of 350 pixels. Using a bigger value in method attributes leads to capping to maximum value and returning an error message.
Sample code to call the method.

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
<script type="text/javascript">
function setHeight(newHeight) {
  svcMca.tlb.api.setToolbarDimensions("MAIN_HORIZONTAL", newHeight, 0, function (response) {
    if (response.result == 'success') {
      alert('Success! New height set!');
    } else {
      alert('Operation finished with error: ' + response.error);
    }
  });
</script>
</head>
<body>
<input type="button" value="Change height" onclick="setHeight(50)"/>
</body>
</html>
```

### Event Listener API: Explained

The event listener API is designed to allow the toolbar register listeners to events that are triggered by Oracle Engagement Cloud. This allows Oracle Engagement Cloud to initiate outbound interaction events and send updates to the toolbar.

The following table lists the usage of the event listener APIs.

<table>
<thead>
<tr>
<th>Method</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>agentStateEvent</td>
<td>Notifies Oracle Engagement Cloud of a change in the user’s signed in or availability status for the specified channel.</td>
</tr>
<tr>
<td>onDataUpdated</td>
<td>Registers a callback for customer data update events to be transmitted to the media toolbar with the updated information.</td>
</tr>
<tr>
<td>focusInteractionEvent</td>
<td>Notifies Oracle Engagement Cloud that the toolbar has changed the previously active interaction slot to a different interaction slot.</td>
</tr>
<tr>
<td>onOutgoingEvent</td>
<td>Registers a callback for the start of an outgoing event triggered from Oracle Engagement Cloud. The callback is initiated by the media toolbar.</td>
</tr>
<tr>
<td>outboundCommError</td>
<td>Notifies Oracle Engagement Cloud that an error occurred while an outbound event is initiated.</td>
</tr>
<tr>
<td>onToolbarAgentCommand</td>
<td>Registers a listener with Oracle Engagement Cloud to provide agent control functionality.</td>
</tr>
<tr>
<td>onToolbarInteractionCommand</td>
<td>Registers a listener with Oracle Engagement Cloud to provide interaction control functionality.</td>
</tr>
</tbody>
</table>
agentStateEvent(channel, eventId, isAvailable, isLoggedIn, stateCd, stateDisplayString, reasonCd, reasonDisplayString, inData, callback, channelType)
Notifies Oracle Engagement Cloud of a change in the user’s signed in or availability status for the specified channel.

The method details are as follows:

- **Parameters**
  - channel: Name of the channel.
  - eventId: Unique identifier of the event.
  - isAvailable: Boolean to indicate if the user is available to receive routed assignments for the specified channel.
  - isLoggedIn: Boolean to indicate if the user is signed in to the specified channel.
  - stateCd: System code to indicate the current availability status of the user.
  - stateDisplayString: Translatable string to display on the Oracle Engagement Cloud user interface that indicates the availability status of the user.
  - reasonCd: (Optional) System code to indicate the reason for the current availability status.
  - reasonDisplayString: (Optional) Translatable string to display on the Oracle Engagement Cloud user interface that indicates the reason for the current status.
  - inData: Object containing the name value pairs if additional details are required.
  - callback: Function to use on the toolbar by Oracle Engagement Cloud to indicate event receipt by Oracle Engagement Cloud.
  - channelType: The type of channel.

Sample code to call the method:

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>
<body>
<input type="button" value="Fire agentStateEvent" onclick="agentStateEvent()"/>
</body>
</html>
```

onDataUpdated (channel, appClassification, callback, channelType)
Toolbar calls this method to register callback listener for any data update that occurs on Oracle Engagement Cloud, and must be transmitted to the toolbar. Using this listener Oracle Engagement Cloud can send notification about contact information being updated in Oracle Engagement Cloud and that the information must be reflected in the toolbar for communication...
snippet. Also, this method can be used by Oracle Engagement Cloud to notify the toolbar of completion of actions such as Wrap-up on Oracle Engagement Cloud.

The method details are as follows:

- **Parameters**
  - **channel**: Name of the channel for which the method is called.
  - **appClassification**: Name of the application classification defined for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of ready to use values, see Application Classification Code: Explained.
  - **callback**: Method callback to let the caller toolbar know of the results.
  - **channelType**: The type of channel for which the method is being called. If the channel is PHONE and the channelType is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.

- The response object has following member attributes
  - **updateType**: Type of update. The values can be CUSTOMER_DATA or WRAPUP_CLOSED.
  - **eventId**: Media event identifier that is used to identify which communication event is targeted by this update.
  - **outData**: Object that has attributes with the names of system token, such as SVCMCA_CONTACT_ID, SVCMCA_CONTACT_NAME, SVCMCA_ORG_ID, and SVCMCA_ORG_NAME.

- Returns: None

Sample code to call the method:

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>
<body>
<input type="button" value="Register data update listener" onclick="onDataUpdated()"/>
</body>
</html>
```

focusInteractionEvent (eventId, channel, channelType, inData, callback)

Notifies Oracle Engagement Cloud that the toolbar has changed the previously active interaction slot to a different interaction slot. This is optional.

- **Parameters**
  - **eventId**: Unique media item identifier. The value must match the supplied by the toolbar in the newCommEvent.
  - **channel**: Name of the channel.
Implementing Service in Engagement Cloud

Chapter 9

Understanding CTI Media Toolbar APIs

Sample code to call the method:

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
<script type="text/javascript">
function focusInteractionEvent() {
svcMcaTlb.api.focusInteractionEvent("1","PHONE","ORA_SVC_PHONE",{},function(response){
if (response.result == 'success') {
alert("success response for focusInteractionEvent received");
} else {
alert("error response for focusInteractionEvent received: "+response.error);
}
},);
}
</script>
<body>
<input type="button" value="Fire focusInteractionEvent" onclick="focusInteractionEvent()"/>
</body>
</html>
```

onOutgoingEvent (channel, appClassification, callback, channelType)

Toolbar calls this method to register callback listener for Oracle Engagement Cloud to trigger an outgoing communication event. Using this callback Oracle Engagement Cloud passes the required information for the toolbar to perform the outgoing event.

The method details are as follows:

- **Parameters**
  - channel: Name of the channel for which the method is called.
  - appClassification: Name of the application classification defined for the current toolbar. If this parameter is passed as null, system assumes that there are no application classification. For a list of ready to use values, see Application Classification Code: Explained.
  - callback: Method callback to let the caller toolbar know of the results.
  - channelType: The type of channel for which the method is being called. If the channel is PHONE and the channelType is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.

- The response passed in the callback has attributes with the names of system tokens, such as SVCMA_ANI, SVCMA_EMAIL, SVCMA_CONTACT_ID, and SVCMA_CONTACT_NAME.

- Returns: None

Sample code to call the method:

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
<script type="text/javascript">
function onOutgoingEvent() {
```
outboundCommError (channel, commUuid, errorCode, errorMsg, callback, channelType)

Toolbar calls this method to notify Oracle Engagement Cloud that an error occurred while initiating the outbound event. The error can occur if the identifier of the event, such as phone number, or email can't be used to establish a connection.

The method details are as follows:

- **Parameters**
  - channel: Name of the channel for which the method is called.
  - commUuid: Unique identifier that is sent with the outgoing event notification from Oracle Engagement Cloud to the toolbar.
  - errorCode: Code to identify the exception that can be used by the agent to identify the nature of the error to contact toolbar support.
  - errorMsg: Message that illustrates the error to the user.
  - callback: Method callback to let the caller toolbar know of the results.
  - channelType: The type of channel for which the method is being called. If the channel is PHONE and the channelType is not provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data: Explained.

- **The response object has following member attributes**
  - result: Represents the status of the running of the method on the server side. This attribute contains success or error.
  - error: Contains error message if the result value is error, else the value is undefined. For more information about error message codes, see Error messages.

- **Returns**: None

Sample code to call the method:

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
<script type="text/javascript">
function onTestError() {
    svcMca.tlb.api.outboundCommError('PHONE', '123456789', '233', 'Could not establish connection, bad number!', function (response) {
        alert('Error message delivered, status: ' + response.result);
    }, 'ORA_SVC_PHONE');
}
</script>
</head>
<body>
<input type="button" value="Register data update listener" onclick="onTestError()"/>
</body>
</html>
```
onToolbarAgentCommand (channel, channelType, executor)

Registers a listener with Oracle Engagement Cloud to provide agent control functionality. This should be called once during initial initialization.

The method details are as follows:

- **Parameters**
  - `channel`: Name of the channel.
  - `channelType`: Type of channel.
  - `executor`: Function implemented by the toolbar will be used with an `agentCommandPrototype` object.

- The `agentCommandPrototype` object has following attributes:
  - `eventId`: unique identifier generated for this command invocation;
  - `command`: name of the command to execute. The following commands are supported:
    - `getCurrentAgentState`
    - `getActiveEngagements`: The `outData` must contain the `activeCount` number and engagements array. For example: `{ activeCount:1, engagements:[ {eventId:"1234"} ] }`. If there are no active engagements, the `engagements` parameter should be an empty array.

  - **Note:** The engagements objects must contain at least an `eventId` attribute to identify the engagement.
    - `makeAvailable`
    - `makeUnavailable`
      - `channel`: Name of the channel.
      - `channelType`: Type of channel.
      - `inData`: Object containing the name value pairs for the command parameters.
      - `result`: Populated by the toolbar upon completion of the command processing with a status of `success` if the command was successfully executed or a failure reason.
      - `resultDisplayString`: Populated by the toolbar for displaying an error to the user.
      - `outData`: Populated by the toolbar if the command requires output data.
      - `sendResponse`: Callback function to use upon completion of the command processing. The `sendResponse` method should be passed the command object with the result populated. Also the `outData` and `resultDisplayString` attributes should be populated if required.

Sample code to call the method:

```html
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js"></script>
</head>
<script type="text/javascript">
function agentCommandExecutor(command) {
  var cmd = command.command;
  switch(cmd) {
  case "getCurrentAgentState":
    command.outData = {
```
function registerAgentCommandListener() {
    svcMcaTlb.api.onToolbarAgentCommand("PHONE", "ORA_SVC_PHONE", agentCommandExecutor);
}
</script>
<body>
<input type="button" value="Register interaction command listener" onclick="registerAgentCommandListener()"/>
</body>
</html>

**onToolbarInteractionCommand(executor)**

Registers a listener with Oracle Engagement Cloud to provide interaction control functionality. This must be called once during initialization. The `executor` function will be used with an interactionCommandPrototype object.

The interactionCommandPrototype has the following attributes:

- **eventId**: Unique media item identifier supplied by the toolbar in the newCommEvent.
- **command**: Name of the command to execute. Currently supported commands are:
  - accept: Accepts an incoming engagement.
  - reject: Rejects an incoming engagement.
  - setActive: Discloses the specified engagement slot on the toolbar.
- **slot**: (Optional) Toolbar slot identifier supplied by the Toolbar in the newCommEvent.
- **inData**: Object containing the name value pairs for the command parameters if required.
- **result**: Populated by the toolbar upon completion of the command processing with a status of success if the command was successfully executed or a failure reason.
- **resultDisplayString**: Populated by the toolbar for displaying an error to the user.
- **outData**: Populated by the toolbar if the command requires output data.
- **sendResponse**: Callback function to use upon completion of the command processing. The sendResponse method should be passed the command object with the result populated. Also the outData and resultDisplayString attributes should be populated if required.

Sample code to call the method:

<html>

```javascript
'channel':command.channel,
'channelType':command.channelType,
'isAvailable':true,
'isLoggedIn':true,
'state':"AVAILABLE",
'stateDisplayString':"Available",
'reason':null,
'reasonDisplayString':null};
break;
case "getActiveEngagements":
    command.outData = {'activeCount':1,'engagements' : [ {eventId:"1234"} ]};
break;
case "makeAvailable":
    alert("makeAvailable command invoked");
    break;
case "makeUnavailable":
    alert("makeUnavailable command invoked");
    }
    command.result = 'success';
    command.sendResponse(command);
}

function registerAgentCommandListener() {
    svcMcaTlb.api.onToolbarAgentCommand("PHONE", "ORA_SVC_PHONE", agentCommandExecutor);
}
</script>
<body>
<input type="button" value="Register interaction command listener" onclick="registerAgentCommandListener()"/>
</body>
</html>

Registers a listener with Oracle Engagement Cloud to provide interaction control functionality. This must be called once during initialization. The `executor` function will be used with an interactionCommandPrototype object.

The interactionCommandPrototype has the following attributes:

- **eventId**: Unique media item identifier supplied by the toolbar in the newCommEvent.
- **command**: Name of the command to execute. Currently supported commands are:
  - accept: Accepts an incoming engagement.
  - reject: Rejects an incoming engagement.
  - setActive: Discloses the specified engagement slot on the toolbar.
- **slot**: (Optional) Toolbar slot identifier supplied by the Toolbar in the newCommEvent.
- **inData**: Object containing the name value pairs for the command parameters if required.
- **result**: Populated by the toolbar upon completion of the command processing with a status of success if the command was successfully executed or a failure reason.
- **resultDisplayString**: Populated by the toolbar for displaying an error to the user.
- **outData**: Populated by the toolbar if the command requires output data.
- **sendResponse**: Callback function to use upon completion of the command processing. The sendResponse method should be passed the command object with the result populated. Also the outData and resultDisplayString attributes should be populated if required.

Sample code to call the method:

<html>

```javascript
'channel':command.channel,
'channelType':command.channelType,
'isAvailable':true,
'isLoggedIn':true,
'state':"AVAILABLE",
'stateDisplayString':"Available",
'reason':null,
'reasonDisplayString':null};
break;
case "getActiveEngagements":
    command.outData = {'activeCount':1,'engagements' : [ {eventId:"1234"} ]};
break;
case "makeAvailable":
    alert("makeAvailable command invoked");
    break;
case "makeUnavailable":
    alert("makeUnavailable command invoked");
    }
    command.result = 'success';
    command.sendResponse(command);
}

function registerAgentCommandListener() {
    svcMcaTlb.api.onToolbarAgentCommand("PHONE", "ORA_SVC_PHONE", agentCommandExecutor);
}
</script>
<body>
<input type="button" value="Register interaction command listener" onclick="registerAgentCommandListener()"/>
</body>
</html>

Registers a listener with Oracle Engagement Cloud to provide interaction control functionality. This must be called once during initialization. The `executor` function will be used with an interactionCommandPrototype object.

The interactionCommandPrototype has the following attributes:

- **eventId**: Unique media item identifier supplied by the toolbar in the newCommEvent.
- **command**: Name of the command to execute. Currently supported commands are:
  - accept: Accepts an incoming engagement.
  - reject: Rejects an incoming engagement.
  - setActive: Discloses the specified engagement slot on the toolbar.
- **slot**: (Optional) Toolbar slot identifier supplied by the Toolbar in the newCommEvent.
- **inData**: Object containing the name value pairs for the command parameters if required.
- **result**: Populated by the toolbar upon completion of the command processing with a status of success if the command was successfully executed or a failure reason.
- **resultDisplayString**: Populated by the toolbar for displaying an error to the user.
- **outData**: Populated by the toolbar if the command requires output data.
- **sendResponse**: Callback function to use upon completion of the command processing. The sendResponse method should be passed the command object with the result populated. Also the outData and resultDisplayString attributes should be populated if required.

Sample code to call the method:

<html>
API Features: Explained

Features are functionality or sets of functionality that are supported by Oracle Engagement Cloud, and can be implemented by a toolbar. Features are supported by API and underlying Oracle Engagement Cloud implementation. List of supported features for version 1 of API is specified in the following table.

<table>
<thead>
<tr>
<th>Feature Code</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INBOUND_CALL</td>
<td>Handles messages related to inbound calls.</td>
</tr>
<tr>
<td>OUTBOUND_CALL</td>
<td>Handles messages related to outbound calls. This API notifies the toolbar to initiate the call using the supplied information.</td>
</tr>
<tr>
<td>TRANSFER_CALL</td>
<td>Handles messages related to a call transfer.</td>
</tr>
<tr>
<td>CONFERENCE_CALL</td>
<td>Handles messages related to a call conference.</td>
</tr>
<tr>
<td>OUTBOUND_CALL_EXTENSIONS</td>
<td>Handles messages related to outbound calls. This API can notify the toolbar to initiate the call with the specified phone number and extension.</td>
</tr>
</tbody>
</table>
System Tokens: Explained

System tokens include the following tokens.

- Data tokens: Represent data values in the Multichannel Application (MCA)
- Control tokens: Control the flow of an interaction

The following table lists the data tokens.

<table>
<thead>
<tr>
<th>Token Name</th>
<th>Token Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service request number</td>
<td>SVCMCA_SR_NUM</td>
<td>Service Request Number to be passed whenever the number is available.</td>
</tr>
<tr>
<td>Contact identifier</td>
<td>SVCMCA_CONTACT_ID</td>
<td>Contact person identifier. The ID may be present on the customer documentation or it can be retrieved from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Account identifier</td>
<td>SVCMCA_ORG_ID</td>
<td>Account or organization identifier. The ID may be present on the customer documentation. The ID can be identified by Interactive Voice Response (IVR) or can be retrieved from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Phone number</td>
<td>SVCMCA_ANI</td>
<td>Contact phone number. Can be obtained from IVR or from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Contact email</td>
<td>SVCMCA_EMAIL</td>
<td>Contact email. Can be obtained from IVR or from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Contact name</td>
<td>SVCMCA_CONTACT_NAME</td>
<td>Contact name. Obtained from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Account name</td>
<td>SVCMCA_ORG_NAME</td>
<td>Account name. Obtained from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Service Request identifier</td>
<td>SVCMCA_SR_ID</td>
<td>Service request identifier. Obtained from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Service Request Title</td>
<td>SVCMCA_SR_TITLE</td>
<td>Service request title. Obtained from Oracle Engagement Cloud for use.</td>
</tr>
<tr>
<td>Communication direction</td>
<td>SVCMCA_COMMUNICATION_DIRECTION</td>
<td>Communication direction, which can be inbound or outbound.</td>
</tr>
<tr>
<td>Interaction identifier</td>
<td>SVCMCA_INTERACTION_ID</td>
<td>Interaction identifier. Obtained from Oracle Engagement Cloud for use. Can be sent from Oracle Engagement Cloud using the</td>
</tr>
</tbody>
</table>
### Control Tokens

The following table lists the control tokens.

<table>
<thead>
<tr>
<th>Token Name</th>
<th>Token Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip customer identification step</td>
<td>SVCMCA_BYPASS_IDENTIFY_CONTACT</td>
<td>Skips automatic customer identification upon receiving an inbound interaction. This token can be used with the startCommEvent() call.</td>
</tr>
<tr>
<td>Skip customer verification step</td>
<td>SVCMCA_BYPASS_CUSTOMER_VERIFICATION</td>
<td>Skips automatic customer verification screen to be presented to agent. This token can be used with the startCommEvent() call.</td>
</tr>
<tr>
<td>Skip the auto screen pop engine</td>
<td>SVCMCA_BYPASS_AUTO_SCREEN_POP</td>
<td>Skips the automatic run of screen pop engine upon accepting the interaction by the agent. This token can be used with the startCommEvent() call.</td>
</tr>
<tr>
<td>Secondary Call Type</td>
<td>SVCMCA_2NDARY_CALL_TYPE</td>
<td>A type of secondary party call, which can be one of Transfer, Conference, and Consult. This value must be sent by the toolbar on newCommEvent() when transferring to the second agent.</td>
</tr>
<tr>
<td>Parent interaction identifier</td>
<td>SVCMCA_PARENT_INTERACTION_ID</td>
<td>Parent interaction identifier in the case of a transfer. Expected to be sent along newCommEvent() data of the second call.</td>
</tr>
</tbody>
</table>

### System Pages: Explained

The following table lists the pages that are available for screen pop. Use the page codes with the `invokeScreenPop()` method to open a screen pop page. For more information about the method, see Interaction API: Explained.

<table>
<thead>
<tr>
<th>Page Name</th>
<th>Page Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Contact</td>
<td>Edit_Contact</td>
<td>Displays details of a contact or customer information. The displayed information can be changed based on the rights associated with agent role.</td>
</tr>
<tr>
<td>Edit Account</td>
<td>Edit_Account</td>
<td>Displays the details of an account or organization information. The displayed page information can be changed based on the rights associated with agent role.</td>
</tr>
<tr>
<td>Page Name</td>
<td>Page Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Edit Service Request</td>
<td>Edit_Service_Request</td>
<td>Shows details of a service request. Page information can be edited according with the rights associated with agent role.</td>
</tr>
<tr>
<td>Create Contact</td>
<td>Create_Contact</td>
<td>Accepts details of a contact.</td>
</tr>
<tr>
<td>Create Account</td>
<td>Create_Account</td>
<td>Accepts details of an account.</td>
</tr>
<tr>
<td>Create Service Request</td>
<td>Create_Service_Request</td>
<td>Accepts details of a service request.</td>
</tr>
</tbody>
</table>

System Business Objects: Explained

The following table lists the business objects.

<table>
<thead>
<tr>
<th>Business Object Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>Describes the contact or customer information.</td>
</tr>
<tr>
<td>Account</td>
<td>Describes account or organization information.</td>
</tr>
<tr>
<td>ServiceRequest</td>
<td>Describes service request information.</td>
</tr>
</tbody>
</table>

Channels Code: Explained

Channels are configured using the Functional Setup Manager within Oracle Engagement Cloud. The following table lists the channel code.

<table>
<thead>
<tr>
<th>Channel Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHONE</td>
<td>Phone channel.</td>
</tr>
<tr>
<td>CHAT</td>
<td>Chat channel.</td>
</tr>
<tr>
<td>EMAIL</td>
<td>Email channel.</td>
</tr>
</tbody>
</table>
Application Classification Code: Explained

The following table lists the out-of-the-box application classification codes that the system recognizes. The list of application classifications can be customized using the Functional Setup Manager.

<table>
<thead>
<tr>
<th>Application Classification Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORA_SERVICE</td>
<td>Default classification for Service related setup for Lookup rules and Screen Pop rules.</td>
</tr>
<tr>
<td>ORA_SALES</td>
<td>Default classification for Sales related setup for Lookup rules and Screen Pop rules.</td>
</tr>
</tbody>
</table>

Error Messages: Explained

The following table lists the codes for error messages that are returned in case of exception while running Oracle Engagement Cloud functionality in response to toolbar requests.

<table>
<thead>
<tr>
<th>Error Message Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVCMCA_ERR_NOTIMPLEMENTED</td>
<td>Specified functionality is not implemented.</td>
</tr>
<tr>
<td>SVCMCA_ERR_MISSING_CONFIGURATION</td>
<td>Specified functionality is based on a configuration information that is missing. For example, call for invokeScreenPop() method with a pageCode attribute that is not present in screen pop pages configuration in Setup and Maintenance.</td>
</tr>
<tr>
<td>SVCMCA_ERR_INVALID_DIMENSIONS</td>
<td>Specified dimension of the toolbars is invalid.</td>
</tr>
<tr>
<td>SVCMCA_ERR_INVALID_CONFIGTYPE</td>
<td>Specified configuration is of an invalid type.</td>
</tr>
<tr>
<td>SVCMCA_ERR_UNKNOWN_METHOD</td>
<td>Unknown method invoked on the API.</td>
</tr>
</tbody>
</table>

Channel Type Data: Explained

The following table lists the channel types, which are predefined lookups identified by the ORA_SVC_CHANNEL_TYPE_CD lookup type.

<table>
<thead>
<tr>
<th>Channel Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORA_SVC_CHAT</td>
<td>Chat channel</td>
</tr>
<tr>
<td>Channel Code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>ORA_SVC_EMAIL</td>
<td>E-mail channel</td>
</tr>
<tr>
<td>ORA_SVC_PHONE</td>
<td>Phone channel</td>
</tr>
<tr>
<td>ORA_SVC_SOCIAL</td>
<td>Social channel</td>
</tr>
<tr>
<td>ORA_SVC_WEB</td>
<td>Web channel</td>
</tr>
<tr>
<td>ORA_SVC_NONE</td>
<td>None</td>
</tr>
</tbody>
</table>
10 Setting Up Service Entitlements

Service Entitlements Overview

Entitlements Background and Terminology

Service entitlements are benefits or privileges you provide to customers when you extend service to them. A milestone is a type of entitlement that identifies service levels your organization has committed to its customers, and is tracked when providing service in the context of a service request.

Oracle Engagement Cloud provides the following ready-to-use milestones:

- **First Response Metric**: Tracks how quickly you agree to respond to a customer after the service request is received.
- **Resolution Metric**: Tracks how quickly you must resolve the service request for the customer.

In addition, Engagement Cloud also enables you to set up administrator-defined milestones that define other service levels you want to track for your customers.

When you create a service request for a customer, milestones are automatically assigned to the service request based on the defined service coverages. Service coverages are implemented using **Standard Coverages**, which act as templates, and **Default Coverages**, which associate the standard coverage templates with a customer, or specify a standard coverage template as a global default that is applicable to all service requests.

Milestone Lifecycle

Once milestones are assigned to a service request, they begin countdown to their respective due dates, and are monitored by the application automatically based on the configuration you have set up.

Depending on your business rules, you can specify whether the milestone should pause countdown (for example, while you are waiting for information from the customer), and when the milestone is considered complete (for example, when the service request is set to **Resolved**).

When a milestone approaches its due date, its status is set to **Warning** by the application, according to the threshold you set up in the standard coverage. You can configure actions to be taken when the milestone reaches the warning threshold (or when it expires) to proactively notify the agent or escalate the service request according to your business needs.

The following topics in this chapter provide details for the setup and configuration of milestones and coverages:

- The topic “Setting Up Milestones and Coverages” provides the details to set up milestones and coverages using the delivered milestones and coverage criteria.
- The topic “Configuring Milestones and Coverages” describes how to expand milestones and coverages to adapt them to your business needs.
Setting Up Milestones and Coverages

Milestones and Coverages: Overview

This section describes how to set up milestones and coverages so that the ready-to-use milestones are automatically applied to service requests and monitored, as described in the previous section.

Setting up milestones to be automatically applied to service requests requires the following steps:

1. Manage milestone configuration (optional)
2. Create a coverage time (optional)
3. Create a standard coverage
4. Apply the standard coverage as global and customer defaults
5. Configure the scheduled process to monitor SR milestones
6. Add current milestone fields to the SR layout (optional)
7. Configure email notifications or other event actions from an object workflow (optional)

Managing Milestone Configuration

The First Response Metric and Resolution Metric milestones are preconfigured for every implementation.

The First Response Metric and Resolution Metric milestones have predefined criteria that specify when the milestone starts, pauses, and completes. You can review these criteria and modify them if required, to match your business rules. To do this, use the Manage Milestone Configuration task in Setup and Maintenance > Service Offering.

For detailed information about managing milestone conditions, as well as creating your own milestones, see the topics "Configuring Additional Condition Columns for Entitlement Rules" and "Setting Up Administrator-defined Milestones".

Creating a Coverage Time

A coverage time is a service calendar containing detailed time intervals that identify when a service request is expected to be worked.

Time outside the specified intervals is skipped when determining the due date and time for a milestone. For example, if your coverage time interval is 9:00 a.m. to 5:00 p.m. Monday through Friday, a milestone that started Friday at 4:00 p.m., which is due in 120 minutes, would be due 10:00 a.m. on Monday.

Coverage times can also contain holidays, which identify days to be skipped when determining the due date. In the previous example, when a milestone started 4:00 p.m. Friday and had 120 minutes until due, if Monday was identified as a holiday, the milestone would be due at 10:00 a.m. on Tuesday instead.

A calendar named 24 by 7 is preconfigured for all implementations. This calendar does not specify holidays or downtime, so the due date for a milestone that uses this calendar, is calculated without skipping any time. To create your own coverage time calendars with the intervals and holidays specific to your business, do the following:

1. Sign in to Engagement Cloud as a user with contract manager duty role. The job roles CRM Administrator and Sales Administrator are both provisioned with this duty role.
2. On the Navigator, click Contracts under Contract Management. The Contracts page is displayed.
3. Click the Tasks icon.
4. In the Setup list of tasks, click Coverage Times. The Manage Coverage Times page is displayed.
5. Click Create to create a new coverage time, or select an existing entry and click Edit to update it.
6. On the Create or Edit Coverage Times page, enter the following details:
   - Name of the coverage time
   - Description
   - Time zone for which you’re creating the coverage time
   - Start and end dates for the coverage time
7. Create a coverage time interval with details of when the coverage time starts, ends, and the days of the week that are part of the interval.
8. Add the holidays that are applicable to the coverage time, and enter the name of the holiday, the start and end date.
9. Click Save.

You can create multiple coverage times and use them as needed in the standard coverage templates. For example, you might have a high severity service request milestone worked using a 24 by 7 calendar, but a lower severity service request milestone might use a different calendar that is only worked during normal business hours. For more details about using coverage times in standard coverages, see the section, “Creating Standard Coverages”.

Creating Standard Coverages

The standard coverage acts as a template that contains entitlement rules, which specify when milestones are due and under what conditions. For example, high severity service requests may have earlier milestone times, and use a calendar with longer work hours and fewer holidays, than lower severity service requests.

To create a standard coverage, do the following:

1. On the Contracts page, click the Tasks icon.
2. In the Setup list of tasks, click Standard Coverage. The Manage Standard Coverage page is displayed.
3. Create a new coverage and select Contracts Service Entitlements as the Entitlement Type. The Edit Standard Coverage page is displayed.
4. Select Actions > Add Entitlements to configure the entitlement rules structure. The Entitlement Rules dialog box is displayed.
   a. Optionally, select Channel Type as an optional condition column, if you want to vary milestones based on the service request channel. Service request severity is automatically included as a condition column.
   b. Click Next, and select all optional results columns to display all available metrics and thresholds.
      You can optionally select only one of the milestones and its threshold if applicable. For example, if service requests that apply to this coverage should not have a first response milestone tracked, you can deselect First Response Metric and First Response Warning Threshold from Optional Results Columns.
   c. Click Finish to apply the entitlement options and complete the configuration of the entitlement rules. The Edit Standard Coverage page is displayed.
5. Add entitlement rules for this coverage.
   a. In the Entitlement Rules section, click the Add Row icon.
   b. Specify the Condition Columns (Severity, and if selected, Channel Type).
      This row is applied when the service request values match the condition column values specified here.
c. Specify the Result Columns (Calendar, along with the metric and threshold values for each milestone).
   
i. From the Calendar drop-down list, select the coverage time calendar to use for calculating the milestones’ due date.
   
ii. For each milestone, enter the metric in minutes, which is used with the calendar to identify when the milestone is due.
   
iii. Enter the corresponding threshold value in minutes. This threshold specifies the number of minutes before expiration that the milestone status is set to Warning.

d. Click Save.

e. Optionally, you can add multiple entitlement rules to a standard coverage.

   Each entitlement rule row is checked against the service request values, and if a service request matches all condition column values, that rule’s milestone results (values and calendar) are used to calculate the milestone due date.

Creating Default Coverages

A default coverage applies a standard coverage template to specific customers, or globally for all customers.

When a service request is evaluated to assign milestones, the application considers all global coverages, and customer default coverages setup for the customer specified on the service request. If a milestone is applicable from both a global default and a customer default coverage, the application uses the customer default coverage values over the global defaults.

To create a default coverage, do the following:

1. On the Contracts page, click the Tasks icon.
2. In the Setup list of tasks, click Default Coverage. The Manage Default Coverage page is displayed.
3. Click Create to add a new coverage.
4. Select the Default Level for the coverage:
   
   o Select Global to assign the coverage to all customers.
      
      Then select the standard coverage in the Coverage column that you want to apply to all customers.
   
   o Select Customer to assign the coverage to a specific customer.
      
      Then select the customer in the Default Level Value column, and then select the standard coverage in the Coverage column that you want to apply to that customer.

5. Enter the start and end date for the coverage.
6. Click Save.

Configuring the Scheduled Process to Monitor SR Milestones

Configure the Monitor Service Request Milestones scheduled process to monitor service request milestone status, setting the job frequency to run the job after specified time interval.

The Monitor Service Request Milestones scheduled process ensures that the service request and milestone status are up-to-date and sends an email notification if compliance issues or warning flags are found. The recommended frequency for running this job is 10 minutes.

For more information about scheduled processes, see Related Topics.
Related Topics

- Profile Options and Scheduled Processes for SR Management

Modifying the Service Request Layout

You can modify the service request layout to show the milestones as a part of the service request details. Fields shown at the service request level represent the next milestone due for the service request. To expose current milestone fields on the service request layout, create a user-defined layout in Setup and Maintenance, and add the MilestoneID field to the Service Request Summary page.

Configuring Email Notifications to Monitor SR Milestones

After you set up the scheduled process, you can configure object workflows to perform actions when a milestone reaches the warning or expired status. For example, you can configure emails to be sent out to warn required teams and managers that a milestone is about to expire, or escalate a service request when a milestone expires.

To configure sending warning emails, do the following.

1. Sign in to Engagement Cloud as an administrator.
2. Navigate to Application Composer and then click Email Templates under Common Setup. The Email Templates page is displayed.
3. Click Create to create a new template. The Create Email Template page is displayed.
4. From the Object drop-down list, select Milestone.
5. Specify the Name, Email Subject, and Email Body for the content of the email you want to send.
6. Click Save and Close.
7. After you create the template, click Settings and Actions on the toolbar, and then click Manage Sandboxes.
8. Select and enter a sandbox.
9. In Application Composer select CRM Cloud in the Application drop-down list, and then select Service as the Object Tags option.
10. Click Object Workflows. The Object Workflows page is displayed.
11. Click Create to create a new object workflow. The Create Object Workflow page is displayed.
12. From the Object drop-down list, select Milestone.
13. Enter the Name and Description for the object workflow.
14. Select the Event Point and Condition for the milestone object workflow.
15. Select the action you want to perform as part of the workflow. You can set up Email Notification and Field Updates for the milestone.
   a. In the Email Notification section, click Create to add a new notification.
   b. Specify the schedule and addresses to which you want to send the notifications.
   c. Click Save to save the milestone object workflow.
16. Republish the sandbox in which you created the object workflow.

You can create a script for specific aspects of the workflow, for example looking up the assignee and assignee manager details for sending the notification. The following is an example script you can use to get the assignee and assignee manager details and populate the Address field for the email notification.

```java
def resourceVO = newView('Resource')
resourceVO.appendViewCriteria("PartyId = ${AssigneePartyId}"") //party id of SR assignee
resourceVO.executeQuery()
def mgrPartyId
```
Configuring Milestones and Coverages

Configuring Milestones and Coverages: Overview

This section describes your options to expand the standard process described in the previous section, to cover a wider variety of scenarios and to adapt milestone functionality to your specific business needs.

Configuring milestones and coverages may include any of the following steps:

- Configuring additional condition columns for standard coverage entitlement rules
- Setting up administrator-defined milestones
- Configuring milestones for multiple business units

Configuring Additional Condition Columns for Standard Coverage Entitlement Rules

Engagement Cloud provides a default entitlement type called Contracts Service Entitlements, which is a matrix class, and a corresponding service mapping that you select when you create your standard coverage.

The Contracts Service Entitlements matrix class defines the structure of the entitlement rules for the standard coverage, including the condition columns (Severity and Channel Type), and result columns (Calendar, and milestone values and warning thresholds). Condition columns in the coverage are compared to field values in a service request to determine which line is applicable, and the result columns display the calendar, milestones, and thresholds that are applied.

You can use other columns from the service request as condition columns in coverages, by editing the service mapping and creating your own matrix class. To do this, first modify the service mapping to add the intended attribute. After you modify the service mapping, create a matrix class including the attributes, and then select the matrix class as the entitlement type for a standard coverage.

The following steps describe how to create a service mapping.

1. On the Contracts page, click the Tasks icon.
2. In the Setup list of tasks, click Manage Service Mappings. The Manage Service Mappings page is displayed.
4. Click the Entities tab, and select Service.
5. In the Service: Details section, click Add Row to add your attribute and select the details, for the mapping.

Note: All attribute names must have the suffix _Custom.

6. Click Save.
9. On the Attribute Mappings tab of the Service: Details section, click Add Row to create the mapping for the new attribute.
10. Select the attribute that you created in step 5 from the Attribute drop-down list and enter the related View Object Attribute of the service request.
11. Click Save.
14. In the Service: Entities tab, click Add Row to create the mapping for the new attribute.
15. Select the attribute that you added in Step 5 from the Attribute drop-down list and enter the values.
16. Click Save and Close.

The following steps describe how to create a matrix class and add additional conditional columns.

1. On the Contracts page, click the Tasks icon.
2. In the Setup list of tasks, click Manage Matrix Classes. The Manage Matrix Classes page is displayed.
3. Create, duplicate, or edit an existing matrix class.
4. On the Edit Matrix Class page for the selected matrix class, in the Condition Columns section, click Add Row to add a new condition column for the mapping that you created in the previous service mapping procedure. This condition column displays in the entitlement rules along with the default columns.
5. Enter the Name of the column and the Source Code Name.
6. In the Compare to Attribute column, select the attribute that you created in the previous service mapping procedure.
7. Click Save and Close.

The matrix class displays under the Entitlement Type drop-down list when you create a standard coverage.

Setting Up Administrator-Defined Milestones

In addition to the predefined milestones for First Response and Resolution, Engagement Cloud enables you to define additional milestones to be tracked against service requests. This enables you to effectively configure the application based on your own business processes and service standards.

The section "Configuring Additional Condition Columns for Standard Coverage Entitlement Rules" described how you can use other columns from the service request as condition columns in coverages, by editing the service mapping and creating your own matrix class. You can also add result columns to a matrix class to represent the administrator-defined milestones. In addition to adding new result columns to the matrix class, you must also do the following:

1. Modify the algorithm that specifies the entitlements to include the new columns you add.
2. Add the new milestone to the Manage Service Milestone Configuration task in Setup and Maintenance.

To add the result columns to the matrix class, do the following:

1. On the Contracts page, click the Tasks icon.
2. In the Setup list of tasks, click Manage Matrix Classes. The Manage Matrix Classes page is displayed.
3. Duplicate the Contracts Service Entitlements matrix class, or edit a matrix class you previously created.
4. On the Edit Matrix Class page for the selected matrix class, in the Result Columns section, click Add Row to add the administrator-defined metric and its corresponding warning threshold.

5. Enter names for the administrator-defined metric and its corresponding warning threshold. The codes are automatically applied in the Source Code Name field.

6. Copy and save the Name and Source Code Name values, because you need Source Code Name in the Manage Algorithms page and Name in the Manage Service Milestone Configuration task page.

7. Click Save and Close.

To modify the algorithm to include the new milestone and threshold columns you created, do the following:

1. On the Contracts page, click the Tasks icon.

2. In the Setup list of tasks, click Manage Algorithms. The Manage Algorithms page is displayed.

3. Select the Contracts Get Service Entitlements row, with the algorithm version 1 and status Published.

4. Select Actions > Create Version. A new version of the algorithm is created, with the status In Progress.

5. Select the new version of the algorithm. The Edit Algorithm page is displayed.

6. On the Algorithm tab, select Look up matching Entitlement Rule.

7. Under First Row Actions, click in the Actions field.

8. Copy the existing algorithm and append it. For example:

   ```java
   if(Matrix.hasProperty("NextResponseMetricCode")) {
     rule= EntitlementResults.insert([EntitlementResultId: getNextId()])
     rule.EntitlementId=Entitlement.EntitlementId
     rule.CalendarId=Matrix.CalendarCode
     rule.ResultName= 'NextResponseMetricCode'
     rule.ResultValue= Matrix.NextResponseMetricCode
     rule.ResultThresholdValue=Matrix.NextResponseWarningMetricCode
   }
   ```

9. Replace the Source Code Name values with the values of the administrator-defined metric and its corresponding warning threshold you saved in the previous procedure for adding the result columns to the matrix class.

   In this example, NextResponseMetricCode and NextResponseWarningMetricCode are the names defined in the matrix classes.

10. Click Save and Close.

11. Navigate to the Manage Algorithms page.

12. Select the modified version of the algorithm, and click Actions > Publish. The Status of the algorithm is updated to Published.

To add the milestone to the Manage Service Milestone Configuration task, do the following:

1. Navigate to Setup and Maintenance.

2. Select the Service offering.

3. In the Functional Areas section, select Service Entitlements.

4. Click Manage Service Milestone Configuration. The Manage Milestone Configuration page is displayed.

5. Click Create Milestone and specify the following milestone details:

   a. Milestone Label: Enter the name of the milestone that should be displayed on the service request.

   b. Milestone Code: Select the name of the milestone result column you added to the matrix class. These names are defined in the matrix classes and cannot be modified.

   c. Milestone Type: Select from the following types, which determine whether milestones are reopened or repeated once complete.

      a. Can be reopened: A milestone that can be reopened is set back to In Progress status when the complete criteria is no longer true, and continues the countdown to expiration when reopened. The Resolution Metric milestone is of this type.
• **Can be repeated**: A milestone that can be repeated cannot be reopened once closed, but another one is created when the start criteria is valid. For example, suppose you configure a Next Response Due milestone that should be repeated each time a customer update is posted to the service request. Once a response is sent, the milestone is complete, but upon receiving a new customer update, another Next Response Due milestone is created and tracked.

• **Cannot be repeated**: A milestone of this type cannot be reopened or repeated. The First Response Metric milestone is of this type.
  
  - **Threshold Code**: Select the name of the threshold result column corresponding to the milestone, which you added to the matrix class. This determines the threshold value that is applied to the milestone. The names are defined in the matrix classes and cannot be modified.
  
  - **Business Unit Name**: Leave as the default business unit (BU) unless you are configuring milestones for multiple business units. For more information, see the next section, “Configuring Milestones for Multiple Business Units”.

6. Enter the milestone conditions. Each milestone (both prebuilt and administrator-defined) has conditions that determine when the milestone starts, pauses, and completes. For each condition, you can define multiple attribute and value pairs that determine when the condition is applied to the milestone.

  - Each tab has a text box that enables you to provide a common description that is displayed in the service request details when the milestone is applied to the service request. This description enables an agent to understand when a milestone starts counting, when it pauses, and when it is considered complete.
  
  - Add conditions using the Create button in the tab. Use the Attribute column to indicate a field to compare to a value using the operator. Alternately, you can specify a static condition that does not require an operator or value.

    - For example, if you specify the field Problem Type Code, you must specify the operator (Is One Of, Is Not One Of) and the values that make this criteria true.
    
    - Alternately, if you specify a static condition such as Immediate, no operator or value is required. In this case, the criteria is considered true as soon as the service request is evaluated.
  
  - When multiple rows are added as conditions, the results are combined logically using the AND operator, so all rows must evaluate to true for the criteria to be true.

### Configuring Milestones for Multiple Business Units

As an administrator, you can create and manage milestones for service requests using the Manage Service Milestone Configuration task in Setup and Maintenance.

When the Business Units feature of Service Offering is enabled, ready-to-use milestones are associated with the default business unit (BU). Milestones are applied only if a BU is defined and associated with the service request, and the milestones must be defined for that BU. If a service request is assigned to a BU, but no milestones are defined for that BU, then the milestones are not applied. If the service request is not associated with any BU, and a default BU is defined for that organization, then the milestones defined for the default BU are applied. If you want to apply a milestone to a new BU, you can create a new milestone configuration, and specify the existing milestone code with the new BU. This enables different business units to have different start, pause, and stop criteria for the same milestone.

You can also define milestones for a specific BU using the Manage Service Milestone Configuration for Business Unit task in Setup and Maintenance. When you use this task, you must specify the BU to set up (Scope) using the Select Scope dialog box. As you create each milestone, the BU specified in the Scope is automatically set as the milestone BU.
11 Setting Up Action Plans

Action Plan: Overview

Action plans can be associated with service requests to help facilitate a series of steps or a sequence of events that may be required to resolve SRs.

Agents can attach action plans to an SR using an administrator-defined template, or by adding individual actions. Actions can be tasks, activities, or appointments, and can be required or optional.

Action plans also provide:

- A visual aid to see the progress of an action plan
- Warnings for both the action plan and individual actions

To use action plans with service requests, you can set up actions, templates, and categories in Functional Setup Manager.

In FSM, select **Service** from the setup choice list. Action Plan appears as an option in the **Functional Area** list. If Action Plan is not displayed, use the **Change Feature opt in** to activate Action Plans.

The following table shows the name and description of each FSM task and whether the task is required.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Action Plan Profile Options</td>
<td>Manage profile options for action plans.</td>
<td>Yes</td>
</tr>
<tr>
<td>Manage Action Plan Actions</td>
<td>Create actions used in action templates. Actions can also be added as individual actions in SRs. You can specify the category, type, duration, and visibility for each action.</td>
<td>Yes</td>
</tr>
<tr>
<td>Manage Action Plan Templates</td>
<td>Create templates to group similar actions to be added to an Action Plan.</td>
<td>Yes</td>
</tr>
<tr>
<td>Manage Action Categories for Action Plans</td>
<td>Action categories are used to facilitate finding actions to add to templates or directly to an action plan.</td>
<td>No</td>
</tr>
<tr>
<td>Manage Template Categories for Action Plans</td>
<td>Template categories are used by agents to filter and find templates to apply to an action plan.</td>
<td>No</td>
</tr>
<tr>
<td>Manage Mapping of Action Plan Status Values</td>
<td>Manage the global setup for status configuration. You can map task, appointment, or service request attributes to an action status at a global level. Action status mapping can also be done at the individual action level.</td>
<td>No</td>
</tr>
</tbody>
</table>
Managing Action Plan Actions

The Managing Action Plan Action task is used to create and edit actions for action templates or individual actions used in SRs. For each action, you specify the category, type, duration, and visibility.

To manage action plan actions:

1. Sign in as an administrator or setup user.
2. Navigate to Setup and Maintenance.
3. Select Service from the Setup drop-down list.
5. Select the Manage Action Plan Actions task.

Creating a New Action

To create a new action:

1. Click Create Action.
2. Enter the action name.
3. Select Category.
4. Select the Action Type.

Action types include:

   o Task
   o Service Request
   o Appointment

5. Select the Action Visibility.

Visibility options include:

   o Not Published - Only visible to the administrator. The action is not available to add to a template and cannot be added by an agent as an additional action.
   o Customer Visible - Visible to the customer when viewing the SR.
   o Internal Only - Visible internally only.

6. Enter a numeric value for how long the task should take and select a unit of measure for the numeric value (days, hours, or minutes).
7. Enter a description of the action and any pertinent details.
8. If necessary, make edits on the Attribute Mapping or Status Mapping tabs, which are explained following these instructions.
9. Click Save and Close.
Attribute Mapping Tab

Every action in an action plan has a related object (as in a task, appointment or service request) that is automatically created when the action starts. The attribute mapping tab is the setup to map information from the action, parent service request, and user-defined values into the related object when it is created.

Attributes to be mapped come from the fields in the related object. Required attributes are automatically listed, and optional attributes can be added. The Mapped To columns indicate where the information comes from to populate the field on the related object.

Mandatory fields are indicated as such in the Required column. Additional fields may also be listed that are produced by the program and cannot be changed. For example, Activity Type displays Task if that action type is Task or defaults to Appointment if the action type is Appointment.

To add an attribute mapping.

1. Click the Add icon.
2. Select the field to be populated from the Attribute choice list.
3. In the Mapped To column, select from where the fields must be populated. Choices are: service request, action, or a user-defined value.
4. Depending on the attribute selected in the previous step, for user-defined values, enter free-form text to the box. For selections such as service request or action, select the field from the choice list in the last column.

Status Mapping Tab

The related business objects created for actions can have different user-defined status values. However, they must be mapped into a new set of status codes for an action in an action plan. You can do this at a global level for all actions, or individually for one action if it has unique attributes. When a user updates an action, the status mapping rules determine how the status of an action updates on an action plan. For example, when a task is set to closed or completed, then the action status is completed.

Note: If the administrator does not define the status mapping at the action level, then the default is the global status mapping.

For more information about the global action status configuration, refer to the topic "Managing Mapping of Action Plan Status Values."

You can use the Status Mapping tab on an action to override the global status mappings. For example, if the global rule is set to an action being closed when the status is complete, but instead you want it to be closed when it’s canceled, you can override the global mapping so that the action closes when both conditions are met.

1. Click the Status Mapping tab.
2. Click the Add icon.
3. Select a status from the Status choice list. This becomes the status that ends up on the action. Status options include:
   - Blocked
   - Completed
   - In progress

Note: Apart from these three status values, you should not set up rules for any other status.
4. Select an attribute from the choice list.
5. Select an operator.
   Operators include:
   - Contains
   - Equals
   - Is null
   - Is not null
6. Select a value for the status.

Dependencies Tab
When an action is added to a template, the **Dependencies** tab shows the template name. The administrator uses this to know what templates are impacted by a change to an action. If the action is in a template, you can also view and edit the template from this tab.

> **Note:** Edits made to templates do not affect active action plans that use the template.

Editing an Existing Action
To edit an existing action:

1. In the **Manage Action Plan Templates** task, select the template you want to edit.
2. Edit the action.
3. Click **Save and Close**.

Managing Action Plan Templates
The Manage Action Plan Templates task is used to create or edit templates that include multiple actions that must be completed to close an SR.

To manage action plan templates:

1. Sign in as an administrator or setup user.
2. Navigate to Setup and Maintenance.
3. Select **Service** from the Setup drop-down list.
4. Select the **Action Plan** functional area.
5. Select the **Manage Action Plan Templates** task.

Creating a New Action Plan Template
To create a new action plan template:

1. Click **Create New Template**.
2. Enter a name for the template.
3. Select a category. Categories are available only if you set them up in the Manage Action Categories for Action Plan task.
4. Select the type of business calendar for the duration and end dates of the actions.
5. Enter a start date for the template.
6. Enter an end date for the template (optional).
7. Check Published to publish the template when it’s ready for agents to use.
8. Enter a detailed description for the template.
9. Add actions to the template by clicking the Add icon in the Action Flow region of the page.
10. On the Add Actions to Template page, search for the action to add by entering a name, category, or action type.
11. Click Search.
12. From the search results, add an action by clicking the Add icon for the action you want to add.
13. Search for, and add all the actions you want to place on the template.

Tip:
- You can delete actions from the Actions to Add pane by clicking the delete icon.
- You can add the same action multiple times. For example, if the action is to obtain approval from two managers, add the action twice.
- You can reorder actions using the arrow icons.
- If you can’t find an action, you can create actions. Click Create New Action.

14. When you have added all the actions you want on the template, click Add to Template.

On the Template page, the actions display in a list view. You can also view the template in a graphical view by selecting the graphic icon.

15. In the Prerequisite column, set any prerequisites for each action in the template. Select the action row in the Prerequisites column and click the Add Prerequisites icon.

16. Select the check box next to the actions that you want to make a prerequisite for the action.

Tip:
- For actions that have prerequisites, mouse over the prerequisite to show the names of the prerequisite actions.
- Use the graphical view to see the order of completion for the actions.

17. Actions are automatically set to mandatory in the Mandatory column. Deselect the check box to make an action optional.

If an action is identified as Optional, the agent has the ability to skip the action if it does not apply to her specific action plan. Mandatory actions cannot be skipped.

18. Click Save and Close to complete the template.

Editing an Existing Template

To edit an existing template:

1. In the Manage Action Plan Templates task, select the template you want to edit.
2. Make your edits.
3. Click Save and Close.
Managing Mapping of Action Plan Status Values

The Manage Mapping of Action Plan Status task is used to create or edit global status mappings from related action business objects to an action plan. The global mapping can be overridden at the individual action level. For example, you can set an action status as complete when the related service request status is resolved or closed.

To manage mapping of action plan statuses:

1. Sign in as an administrator or setup user.
2. Navigate to Setup and Maintenance.
3. Select Service from the Setup drop-down list.
5. Select the Manage Mapping of Action Plan Status Values task.

On the Action Status Configuration page, select from the action types, Appointment, Service Request, or Task. You can edit the existing status, or click the Add icon to add additional status configurations.
12 Enabling Productivity Tools

Enabling Productivity Tools for Service Requests: Explained

Productivity tools for service requests display as features on create or edit pages for service requests. The following table lists and describes the productivity tools that are available for service requests.

<table>
<thead>
<tr>
<th>Productivity tool</th>
<th>Description</th>
<th>Profile Option Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SmartText</td>
<td>A SmartText entry is a reusable fragment of text that you can insert in messages and fields</td>
<td>SVC_ENABLE_STD_TEXT_IN_SR</td>
</tr>
<tr>
<td>Keyboard Shortcuts</td>
<td>Keyboard shortcuts are a combination of keyboard keys that can be used to quickly access actions and buttons on the service request pages.</td>
<td>SVC_ENABLE_KEYBOARD_SHORTCUTS_IN_SR</td>
</tr>
<tr>
<td>SR Audit</td>
<td>The SR Audit subtab displays the history of service requests from the time they are created.</td>
<td>SVC_ENABLE_AUDIT_IN_SR</td>
</tr>
<tr>
<td>Severity color coding</td>
<td>You can color code different severity levels for the service requests to quickly identify the severity.</td>
<td>SVC_ENABLE_SEVERITY_COLOR_IN_SR</td>
</tr>
<tr>
<td>Knowledge</td>
<td>Displays the list of knowledge articles within a service request to help agents find and use the appropriate details.</td>
<td>SVC_ENABLE_KNOWLEDGE_IN_SR</td>
</tr>
</tbody>
</table>

The productivity tools are disabled by default. You can enable these features using the following procedure.

1. Sign in to Engagement Cloud as an administrator.
2. From the Navigator, click Setup and Maintenance.
3. Select the Service offering.
4. In the Functional Areas section, select Productivity Tools.
5. Click the Manage Global Search Profile Options task.
6. Search for the profile option code for the productivity tools that you want to add.
7. In the Profile Values section for the profile option code, select Yes in the Profile Value drop-down list.
8. Save the configuration.

After you enable the productivity tools, set up and configure each tool according to your requirements.

Updating the Status List Order

The status list on the Summary Details page contains a list of statuses that can be assigned to an SR. The order of the statuses displayed in this list is dependent on the ranking assigned to each status. You can update the ranking for the statuses in Setup and Maintenance. The following procedure describes how to update the status ranking.

1. Sign in to Engagement Cloud as an administrator.
2. From the Navigator, click **Setup and Maintenance**.
3. Select the **Service**.
4. In the **Functional Areas** section, select **Service Request**.
5. In the **Service Request** tasks pane, select **All Tasks** from the **Show** drop-down list, and then click **Manage Service Request Status Values**.

The Service Request Statuses page is displayed.
6. Update the **Ranking** column to position the status in the status list according to your requirements.
7. Click **Save and Close**.

**Related Topics**
- Adding Service Request Severity Values: Procedure
- Viewing Audit History for a Service Request

### Managing SmartText Entries: Explained

A SmartText entry is a reusable fragment of text that you can insert in messages and fields. As an administrator, you can create public SmartText entries that all users can use in their service requests. Users can’t edit these entries.

You can define where to save the SmartText entry, insert variables in the entry and choose to share the SmartText entry by using the following availability options:

- **Always**: Select this option to make the SmartText entry available to users at all times.
- **Interval**: Select this option and define the period during which this entry is available to users.
- **Disabled**: Select this option to disable the SmartText entry and display it in the public folders for the users.

### Creating a Public SmartText Entry

Public SmartText entries are SmartText entries that are created by the administrator and made available to the users. As an administrator, you can create a public SmartText entry from the Create Service Request or the Edit Service Request pages.

To create a public SmartText, do the following:

1. On the Create or Edit Service Request page, open the SmartText pane by clicking the arrow on the **Restore Pane** arrow.
2. Click the **SmartText** tab.
3. Click the **Manage SmartText** icon.
4. On the **Manage SmartText** page, on the **Public** tab, select a folder in which you want to create the public SmartText entry. You can either select an existing folder, or create a new folder to save your SmartText entry.
5. Click **Create** and select **New SmartText**.
6. On the New SmartText page, enter the following information:
   - Name of the SmartText entry.
   - Select whether you want to enable HTML to use rich text for the content.
   - Enter the content of the SmartText entry.
   - To insert a variable in the text, enter the at sign (@), then enter the variable name. You can press the Tab key to automatically complete the variable, or use the navigation arrows on your keyboard, then press the Enter key. The variable is inserted where the cursor stands in your text.
   - Select **Publish** to save this entry in the **Public** tab for all users. If you don’t select **Publish**, the SmartText entry is saved in your **Private** folder and is available only to you.
o Verify the location where you’re saving the SmartText.
o Select the availability of the SmartText entry from the list of Availability options.

7. Click Publish to publish the SmartText entry.

After you create Public SmartText entries, you can edit and delete the entries. You can also move the entries to different folders, and duplicate a SmartText entry to create a new one.

Managing Keyboard Shortcuts: Explained

As an administrator, you can edit the keyboard shortcuts for users to manage service requests. The keyboard shortcuts can be defined under Setup and Maintenance. You can set keyboard shortcuts for the following:

- Button Access Keys: Button access keys are the buttons and links provided on the service request pages, such as Submit, Done, Apply, Response, Save and Close, and so on. The following table lists the button access key combinations for supported browsers.

<table>
<thead>
<tr>
<th>Browser</th>
<th>Operating System</th>
<th>Key Combination</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Chrome</td>
<td>Linux</td>
<td>Alt+key</td>
<td>Click</td>
</tr>
<tr>
<td>Google Chrome</td>
<td>Mac OS X</td>
<td>Control+Option+key</td>
<td>Click</td>
</tr>
<tr>
<td>Google Chrome</td>
<td>Windows</td>
<td>Alt+key</td>
<td>Click</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>Linux</td>
<td>Alt+Shift+key</td>
<td>Click</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>Mac OS X</td>
<td>Control+key</td>
<td>Click</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>Windows</td>
<td>Alt+Shift+key</td>
<td>Click</td>
</tr>
<tr>
<td>Microsoft Internet Explorer 7</td>
<td>Windows</td>
<td>Alt+key</td>
<td>Set focus</td>
</tr>
<tr>
<td>Microsoft Internet Explorer 8</td>
<td>Windows</td>
<td>Alt+key</td>
<td>Clear or set focus</td>
</tr>
<tr>
<td>Microsoft Internet Explorer 11</td>
<td>Windows</td>
<td>Alt+key</td>
<td>Clear or set focus</td>
</tr>
<tr>
<td>Apple Safari</td>
<td>Windows</td>
<td>Alt+key</td>
<td>Click</td>
</tr>
<tr>
<td>Apple Safari</td>
<td>Mac OS X</td>
<td>Control+Option+key</td>
<td>Click</td>
</tr>
</tbody>
</table>

The following table lists the keyboard shortcuts that are provided with Engagement Cloud. Some of the shortcuts cannot be modified.

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard shortcut</th>
<th>Editable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Service Request</td>
<td>E</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Action Commands

Action commands are task actions that you can perform on the service request, such as create response, add internal note, update the service request milestone, assign a service request to yourself, update and so on. The keyboard shortcut combination, Alt+Control+Key, can be used for all actions on the user interface. This combination is applicable to both Windows and Mac OS X operating systems. The following table lists the keyboard shortcuts provided with Engagement Cloud. All action command shortcuts can be modified.

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard shortcut</th>
<th>Editable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save and Continue</td>
<td>V</td>
<td>Yes</td>
</tr>
<tr>
<td>Send</td>
<td>D</td>
<td>Yes</td>
</tr>
<tr>
<td>Post</td>
<td>T</td>
<td>Yes</td>
</tr>
<tr>
<td>Select from Service Request</td>
<td>Q</td>
<td>Yes</td>
</tr>
<tr>
<td>Insert Knowledge</td>
<td>W</td>
<td>Yes</td>
</tr>
<tr>
<td>Add Team Members</td>
<td>A</td>
<td>Yes</td>
</tr>
<tr>
<td>Back</td>
<td>B</td>
<td>No</td>
</tr>
<tr>
<td>Cancel</td>
<td>C</td>
<td>No</td>
</tr>
<tr>
<td>No</td>
<td>N</td>
<td>No</td>
</tr>
<tr>
<td>Save and Close</td>
<td>S</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Y</td>
<td>No</td>
</tr>
<tr>
<td>Apply</td>
<td>L</td>
<td>No</td>
</tr>
<tr>
<td>Done</td>
<td>P</td>
<td>No</td>
</tr>
<tr>
<td>Finish</td>
<td>I</td>
<td>No</td>
</tr>
<tr>
<td>Next</td>
<td>X</td>
<td>No</td>
</tr>
<tr>
<td>OK</td>
<td>K</td>
<td>No</td>
</tr>
</tbody>
</table>

### Table: Action Commands

<table>
<thead>
<tr>
<th>Action</th>
<th>Keyboard shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Action Plan</td>
<td>A</td>
</tr>
<tr>
<td>Show Keyboard Shortcuts</td>
<td>H</td>
</tr>
</tbody>
</table>
The keyboard shortcuts are provided with default configurations. You cannot create shortcuts, but you can edit some of the existing keyboard shortcuts. To edit keyboard shortcuts, do the following:

1. Sign in to Engagement Cloud as an administrator.
2. Navigate to Setup and Maintenance.
3. Click the Tasks icon and select Search.
4. Search for and select the Manage Service Request Keyboard Shortcuts task.
5. On the Manage Keyboard Shortcuts page, review the keyboard shortcuts set for the actions and the button access keys. Select the keyboard shortcut that you want to change and update the key in the Shortcut Key column. This key is not case-sensitive.

> Note: No two actions or buttons can have the same keyboard shortcut.

6. Click Save.
13 Setting Up Social Networking

Social Networking: Overview

Oracle Social Network is a secure, private social network that integrates with Oracle Sales Cloud and connects you with all your colleagues. With Oracle Social Network, you and your teams have the tools to collaborate, capitalize on collective experience, and make informed business decisions.

You can use Oracle Social Network for:

- Discussing projects, plans, and issues in public forums, membership groups, or one-on-one.
- Reviewing and publishing files.
- Following the daily activities of the people you choose.

The real power of Oracle Social Network is how it integrates with common sales objects. For example, you can bring an Oracle Sales Cloud opportunity into a Conversation where you can discuss it, plan around it, and share it. You can take the opportunity from possibility to realization without losing any of the casual and formal information that flows from all of this activity.

Note: You can create custom reports for Lightbox. For more information see the Oracle Sales Cloud Creating and Administering Analytics for Sales guide.

Enabling Oracle Social Network Objects for Service Requests: Explained

You can configure service requests to be sharable on Oracle Social Network. This feature enables customer service representatives to start conversations, share documents related to service requests, and solicit feedback from other resources in sales, service or elsewhere in the enterprise.

To enable Oracle Social Network for service requests:

1. Sign in as an administrator or setup user.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select Service.
4. Select the Productivity Tools functional area and then select the Manage Oracle Social Network Objects for Service task.

   The Manage Oracle Social Network Objects for Service page appears.
5. Expand the Service name, and then click Service Request.
6. Click the Enable Object button.
7. In the Service Request Enable Object dialog box, select Manual, then click OK.
Note: Although you can select Automatic, doing so means every service request created is shared on Oracle Social Network, which is typically not recommended. Selecting Manual indicates that the user can explicitly share service requests only when collaboration is required.

8. Define the service request attributes that you want to share by clicking the + (plus) button in the Business Object Attributes work area.

9. Click Save.

Oracle Social Network is enabled for service requests.

Note: Once Oracle Social Network is enabled, use your Service Request page layouts in Application Composer to control whether the Social (OSN) subtab is available to specific user roles.

Related Topics

- About Setting Up Oracle Social Network
- Dynamic Page Layouts: Explained
14 Setting Up Service Business Units

Setting Up Business Units for Service: Overview

With Business Units for Service you can deploy more than one service center within a single instance of Engagement Cloud. This topic gives an overview of the steps that must be complete the business units setup.

Multiple business units enables you to:

- Segment SRs between business units so that users can search and identify SRs from multiple BUs.
- Use product catalogs, categories, channels, and email templates specific to a business unit.
- Assign SRs to queues by writing rules based on business unit.
- Create service request BI reports specific to a business unit.

Objects not currently supported by multiple business units in Service are accounts and contacts, users, resources, and lookups.

For more detailed information about business units, refer to the Setting Up Multiple Business Units chapter of the Implementing Sales guide at http://docs.oracle.com/cloud.

For more detailed information about users and security, refer to the Getting Started with Your Sales Implementation guide at http://docs.oracle.com/cloud.

The following table shows the order of tasks that must be set up business units in Service.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting Up Business Units for Service</td>
<td>Add the Manage Business Unit functional area to the Service offering by using the Change Feature opt in in FSM.</td>
</tr>
<tr>
<td>Manage Common Profile Options</td>
<td>Set the profile options to enable the multi-BU functionality in the Manage Common CRM Business Unit Profile Options task.</td>
</tr>
<tr>
<td>Manage Internal Resource Organizations</td>
<td>Define internal resource organizations to be associated with the business unit.</td>
</tr>
<tr>
<td>Manage Resource Organization Hierarchies</td>
<td>Add the internal resource organizations to the internal resource organization hierarchy.</td>
</tr>
<tr>
<td>Create Business Unit</td>
<td>Create a business unit to be associated with the resource organization in the Manage Business Unit Task.</td>
</tr>
<tr>
<td>Associate Resource organization to Business Unit</td>
<td>Associate the internal resource organization to the business unit you created. Use the Resource Directory.</td>
</tr>
<tr>
<td>Create Employees</td>
<td>Add users to the organization in the Users, Roles and Delegations task, and make one of the resources a manager for the organization.</td>
</tr>
<tr>
<td>Reset Passwords for Users</td>
<td>Reset the password for the users.</td>
</tr>
</tbody>
</table>
Oracle Engagement Cloud
Implementing Service in Engagement Cloud

Chapter 14
Setting Up Service Business Units

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the Scope in Service Setup Tasks and complete Service Setup tasks.</td>
<td>Set the scope for Service tasks and setup the remainder of the service offering.</td>
</tr>
<tr>
<td>Add additional BU fields in the SR if users are associated with multiple BUs.</td>
<td>Change the layouts of the SR pages if required. Use Apps Composer to include multiple business unit fields in the SRs.</td>
</tr>
</tbody>
</table>

**Note:** Only required if a user is associated with multiple business units.

### Related Topics

- Getting Started with Your Sales Implementation
- Implementing Sales

### Managing Common CRM BU Profile Options for Service BU

To enable multiple business units (BUs), you must set the two profile options discussed in this topic.

The two profile options are:

1. Multiple Business Units Enabled (HZ_ENABLE_MULTIPLE_BU_CRM): Set this profile option to Yes. The default value is No.
2. Customer Relationship Management Business Unit Default (HZ_DEFAULT_BU_CRM): Set this to the default Engagement Cloud business unit.

Use the following procedure to set the profile options:

1. Sign in as a setup user or administrator.
2. Navigate to the **Setup and Maintenance** work area.
3. Click the Setup drop-down list and select **Service**.
4. Select **Company Profile** in the list of functional areas.
5. Select the **Manage Common CRM Business Unit Profile Options** task.

   The Manage Common CRM Business Unit Profile Options page shows the two profile options.

6. Select the **HZ_ENABLE_MULTIPLE_BU_CRM** profile option and set it to Yes.
7. Click **Save and Close**.
8. Select the **HZ_DEFAULT_BU_CRM** profile option.
9. Click **Save and Close**.

### Managing Internal Resource Organizations for Service BU

In this task, you define internal resource organizations to be associated with the business unit. A resource organization represents the internal organization and structure for the business unit. Resource organizations are hierarchically structured, and the organization hierarchy helps to derive the reporting relationships.
To define the internal resource organization:

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
3. Select the Setup drop-down list and select the Service offering.
4. Select the Users and Security functional area.
5. Select the Manage Internal Resource Organizations task.
6. Click the Add icon to add a new resource organization.
7. Click the Option 2: Create New Organization option to create a new organization.
8. Click Next.
9. On the Create Organization: Enter Basic Information page, enter a Name for the organization.
10. Click the Add icon in the Organization Usages region.
11. Select Resource Organization from the Usage choice list.
12. Click Finish.

Managing Resource Organization Hierarchies for Service BU

In this task, you add the internal resource organizations to the internal resource organization hierarchy. A resource organization hierarchy is a hierarchically structured representation of the way resources are grouped within a resource organization.

To add internal resource organizations to the resource organization hierarchy:

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
3. Click the Setup drop-down list and select the Service offering.
4. Select the Users and Security functional area.
5. Select the Manage Resource Organization Hierarchies task.
7. Click the link for the resource organization you want to edit.
8. Select Edit This Hierarchy Version from the Actions menu.
9. Expand the organization list in the Internal Resource Organization Hierarchy region.
10. Add the organization you created in Manage Internal Resource Organizations to the organization hierarchy by selecting the organization you want to add.
11. Click the Add icon.
12. In the Add Tree Node window, click Search.
13. In the Search Note window, search for the organization you created in the Manage Internal Resource Organizations task.
14. Click OK to add the organization.
   To add more organizations, highlight parent to make child. Repeat the steps to search and add as many times as needed.
15. Click Save and Close.
16. Click Yes on the warning message letting you know that the hierarchy version is to be updated and the corresponding reporting hierarchy regenerated.
Creating a Business Unit for Service BU

In this task, you create a business unit and associate it to the resource organization.

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
3. Select the Setup drop-down list and select the Service offering.
4. Select the Business Units functional area.
5. Select the Manage Business Unit task.
6. On the Create Business Unit page, Click the Add icon.
7. Enter a name for the BU.
8. Click Search in the Default Set choice list.
9. Search for Common in the Reference Data Set Name field.
10. Select COMMON from the search results.
11. Click OK on the Search and Select window.
12. Click Save and Close on the Create Business Unit page.

To add another business unit, select the Manage Business Unit task again and repeat the steps.

Associating Resource Organizations to Business Units for Service BU

In this task, you associate a business unit to the organization. By associating resource organizations with business units, you can control access to the transactional data available to service resources in business objects like service requests.

To associate resource organizations to business units:

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
4. Select the Business Units functional area.
5. Click View Organizations under Resource Organizations.
6. Search for the organization you created.
7. In the Search Results region, select the link for the organization.
8. Select the Business Units tab.
9. Click the Add icon.
10. Select the business unit from the choice list.

This becomes the primary business unit for the resource organization. If you add more, you can change the primary business unit.
11. To add more business units, click Save then the Add icon.
12. When finished Click Save and Close.
13. Click Done.
Creating Employees for a Service BU

In this task, you add resources to the organization. When you add a resource to an organization, the resource becomes a member of the organization and a part of the organization hierarchy.

First, create a manager for the organization. To create application users, use the Users, Roles and Delegations task in the Setup and Maintenance work area:

1. Sign in as an administrator or a setup user.
2. Navigate to Setup and Maintenance.
3. Select Users, Roles and Delegations in the navigator.
4. On the Search Person page, click the Create icon.
5. On the Create User page, enter the Last Name.
6. Enter the First Name.
7. Enter the Email.
8. In the User Details region, enter a User Name.
9. In the Employment Information region, select Employee from the Person Type choice list.
10. Select a Legal Employer from the choice list.
11. Select the Business Unit of the employee from the choice list.

Note: This is the Business unit of the employee and not the business unit of the resource organization. They both may be different. What business unit is selected for the employee information is based on how employees are organized.

12. In the Resource Information region, select the Resource Role from the choice list.
13. Search for and select an organization from the Organization choice list. This is the organization you created earlier and how the agent is associated with the BU.
14. Click Autoprovision Roles. This gives the user any predefined job roles.
15. Click Save and Close.

You can view everyone you created in the Resource Directory. Select View Organization and look on the Members tab.

Related Topics
• Setting Up Users and Security: Overview
• About Security Roles: Explained

Working with the Scope in Service BU Setup

When you opt in to the Business Units feature for Service, the Scope in the task list contains a link to set the scope of the task. This topic covers setting the scope for tasks when setting up business units for Service.
For multiple business units in Service, the setup tasks are the same as in a normal Service setup, except that with Service BU you can use the default site-value profile option, or select the business unit profile value.

To set the scope for tasks:

1. Click the Select link in the Scope column of the task list. If you set the scope previously, the link displays the previous BU that was set as the scope.
2. On the Select Scope window, choose Select and Add from the Business Unit choice list.
3. Click Apply and Go to Task.
4. In the Select and Add: Business Unit window, search for and select the business unit you want to set for the scope.
5. Click Save and Close.

The page opens for the task you’re working with. Here, you can choose to use the default Site Level Value or select the Business Unit Profile Value.

To select a Business Unit Profile Value:
6. Deselect the Use Site Value check box.
7. In the Business Unit Profile Value field, enter the profile value for the business unit.
8. Click Done.

The task closes and now on the Setup page, the business unit you set for the Scope is populated for all tasks. Each task you open now is the setup for the business unit in the scope column.

To set up additional business units, follow the same steps again.

>Note: After you set up the first business unit, the Business Unit drop-down list in the Select Scope window now shows the business units you already set up.

For all tasks, the scope displays the business unit you’re currently working with.

### Completing the Setup Tasks for Service BU

Once the preliminary steps for setting up Service business units are done, you can set up the remainder of the service offering.

The steps for setting up the tasks in Service for business units are the same except with business units; you can either use the site-value profile option, or select the business-unit profile value.

The following table is a list of the other Service optional setup tasks and the help topics that provide more information.

<table>
<thead>
<tr>
<th>Service Task</th>
<th>Related Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Service Product Groups Usage for business unit.</td>
<td>Defining a Catalog for the Service Offering: Explained</td>
</tr>
<tr>
<td>Manage HR Help Desk Product Group Usage for Business Unit</td>
<td></td>
</tr>
<tr>
<td>Manage Service Categories for Business Unit</td>
<td>Managing Service Request Categories: Explained</td>
</tr>
<tr>
<td>Manage HR Help Desk Service Categories for Business Unit</td>
<td>Managing Service Request Categories and Product Usage Groups for HR Help Desk: Explained</td>
</tr>
</tbody>
</table>
### Service Task

<table>
<thead>
<tr>
<th>Manage Communication Channels for Business Unit</th>
<th>Setting Up Communication Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Inbound Email Profile Values for Business Unit</td>
<td>Creating and Updating Inbound Message Filters: Procedure</td>
</tr>
</tbody>
</table>

**Note:** When setting up email profile values for a specific business unit, you must decide whether use the standard profile value or a value specific to a business unit.

<table>
<thead>
<tr>
<th>Manage Service Email Templates for Business Unit</th>
<th>Defining Email Templates: Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage HR Help Desk Email Templates for Business Unit</td>
<td>Configuring HR Help Desk Email: Explained</td>
</tr>
</tbody>
</table>

### Related Topics

- Defining a Catalog for the Service Offering: Explained
- Managing Service Request Categories: Explained
- Creating and Updating Inbound Message Filters: Procedure
- Defining Email Templates: Procedure

### Securing the Business Unit Field on a Service Request

The business unit (BU) field on the service request object is secured by using the Update Service Request Business Unit privilege.

The following prebuilt job roles can change the business unit when they edit a service request:

- Customer Service Representative
- Customer Service Manager
- Sales Manager
- Sales Representative
- Sales VP
- Customer Relationship Management Application Administrator
- Sales Administrator
- Channel Account Manager
- Channel Operations Manager
- Channel Sales Manager
Removing the Update Service Request Business Unit Privilege

If the Update Service Request Business Unit privilege is revoked from the job roles mentioned in the previous list, users with those roles cannot change the BU of the service request when editing.

hawkernote: You can remove the Update Service Request Business Unit privilege from a user’s job role only if you had already granted one of the prebuilt service job roles to each of your users.

To remove the Update Service Request Business Unit privilege from a user’s job role, perform the following steps in Security Console:

1. From the previous list of job roles, copy the prebuilt job role that is granted to the user. This prebuilt job role should have one of the following duty roles:
   - Service Request Administrator
   - Service Request Power User
   - Service Request Troubleshooter
   - Service Request Contributor
   - Service Request Channel User
2. Copy the duty role that is already granted to the copied job role.
3. Edit the copied duty role and remove the Update Service Request Business Unit privilege from the role.
4. Edit the previously copied job role. Remove the service duty role that you copied to create a custom duty role.
5. Add the custom duty role to the copied job role.
6. Remove the previously granted prebuilt service job role from the user.
7. Grant the new custom job role to the user.

If you have custom roles for your users, complete the following steps to remove the Update Service Request Business Unit privilege from the user’s job role:

1. Identify the custom role that has the Update Service Request Business Unit privilege.
2. Remove the Update Service Request Business Unit privilege from the role.

Related Topics

- Copying Sales Roles: Points to Consider
- Reviewing Roles on the Security Console: Overview

Exporting and Importing the Functional Setup Data for Business Units

You can export or import the functional setup data for all business units in the Service offering by using the export and import feature in Functional Setup Manager. To complete this process, you can use an implementation project or the implementation method based on offering.

When you use the method based on offering, all the functional setup data for the Service offering is exported or imported, including the setup data for all the business units. You can also export and import the functional setup data for a specific
business unit by using an implementation project. This feature provides additional management flexibility in scenarios where each business unit manages its own set of configurations.

For more information about using implementation projects, the export and import feature, and the implementation method based on offering in Functional Setup Manager, see the Using Functional Setup Manager guide at the following location: https://docs.oracle.com/en/cloud/saas/applications-common/18b/oafsm/toc.htm

To export the functional setup data for business units in the Service offering by using an implementation project:

1. Click Navigator > Setup and Maintenance work area to open the Setup page.
2. Select Manage Implementation Projects from the task panel to open the Manage Implementation Projects page.
3. Create a new implementation project to export the functional setup data for one of the following:
   - A specific business unit
   - All business units
4. Save and open the implementation project.
5. In the Task Lists and Tasks area, ensure that you add the Define Business Units for Service task list to the project.
6. Determine whether you want to export the setup data for a specific business unit or all business units.
   - If you want to export the setup data for all business units, then go to step 12.
   - If you want to export the setup data for a specific business unit, then complete all the remaining steps.
7. Expand the task list.
8. Navigate to the first task that has a Select link in the Selected Scope column. Click the Select link.
9. From the Business Unit list, select Select and Add.
10. Click Apply and Go to Task.
11. In the Select and Add Business Unit window, search and add the specific business unit. The selected business unit appears in the Selected Scope column for all the tasks in the task list.
12. Click Done for the implementation project.
13. Continue to create the configuration package as described in the "Exporting Setup Data Using Implementation Project" process in the Using Functional Setup Manager guide.

When you export the configuration package:

- If you select all the business units in step 3, the export data includes the functional setup data for all the business units in the Service offering.
- If you select a specific business unit in step 3, the export data includes the functional setup data only for the selected business unit.

Note: To import the functional setup data for business units, follow the steps described in the Using Functional Setup Manager guide.

Related Topics

- Using Functional Setup Manager Guide
- Exporting and Importing Setup Data Using Implementation Project: Explained
- Exporting Setup Data Using Implementation Project: Procedure
- Exporting and Importing Setup Data by Offering or Functional Area: Explained
Setting Up Service Request Visibility Based on BU:
Overview

When users view lists of service requests or create user-defined searches, you can restrict their access based on their business unit (BU) membership. The predefined roles don’t have service request visibility based on business unit. In the Security Console, you can assign BU-based visibility to service requests for specific roles. Consequently, users with these specific roles can see only the service requests assigned to the business units where they’re a resource member.

With this new data security policy, your company has the option to ensure that all predefined and user-defined searches are restricted only to business unit membership for a set of users.

For more information about assigning the data security policies based on business unit to your users, see Setting Up Service Request Visibility Based on BU.

Related Topics

- Setting Up Service Request Visibility Based on BU
Implementing Service Analytics: Explained

Oracle Transactional Business Intelligence (OTBI) is a real time, self-service reporting solution bundled with Oracle Engagement Cloud. It provides prepackaged analytic content built on the Oracle Business Intelligence (BI) platform. These include subject areas for building your own dynamic analyses using an intuitive interface, industry standard metrics, and role-based, best practice reports and dashboards that deliver up to the minute business insight across the entire gamut of your service-related business operations.

You can view the analyses in the following ways:

- To access BI Catalog, select **Navigator > Tools > Reports and Analytics**. The Reports and Analytics page is displayed.
  - Click the **Hierarchical Selector** > icon or the **Browse Catalog** button. The BI Catalog is displayed, where you can view your personal and shared analyses. You can also create analyses based on your requirements using subject areas.

- To access the Analytics page, from the Home page, click the **Service** group icon, and then click the **Analytics** icon, if enabled. For information about enabling the **Analytics** icon for Service, see the "Enabling the Service Analytics Icon on the Springboard" topic in this guide. The Analytics page is displayed. You can search for the analyses available in BI Catalog, and mark your favorites. These favorites persist on the Analytics page as long as they remain favorites. This page also shows the analytics recently viewed by the signed in user.

- To access the Service Infolets page, from the page control on the Home page, click the **Service Infolets** icon. Prebuilt and administrator-defined or user-defined infolets can be displayed on the Service Infolets page.

Managing Service Infolets

For the prebuilt Service roles, the Service Infolets page displays the infolets based on the user’s role. For administrator-defined roles, administrators must enable the Service Infolets page for each new role.

The procedures to create, manage, and enable Service infolets are the same as that for Sales infolets. For more information about creating and managing infolets, see "Modifying Infolets" under Related Topics. For information about the procedure to enable infolets on the dashboard, see "Enabling the Sales Infolet Pages" under Related Topics. For information about building analytics, see "Creating and Editing Analytics" under Related Topics.

**Related Topics**

- About Security Roles: Explained
- Enabling the Sales Infolet Pages
- Creating and Editing Analytics
- Modifying Infolets
Enabling the Service Analytics Icon on the Springboard

On the Home page, the Service group icon does not display the Analytics icon by default. The administrator must enable it explicitly.

To enable the Analytics icon, do the following:

1. Click your user image or name in the global header, and select Manage Sandboxes under the Administration menu. The Manage Sandboxes window is displayed.
2. Create a sandbox or select an existing one, and click Set as Active to activate the sandbox. The sandbox is designated as the active sandbox.
3. Close the Manage Sandboxes window.
4. In the Navigator menu, select Structure. The Navigation Configuration page is displayed.
5. Expand the Service menu and select Analytics. The Edit Page Entry page is displayed.
6. From the Show on Navigator drop-down list, select Yes.
7. From the Show on Springboard drop-down list, select Yes.
8. Click Save and Close
9. Navigate to the Home page and verify that the Analytics icon is displayed under the Service group icon.
10. Publish the sandbox.

Roles for Service Analytics

The Service analyses are visible for customer service representative, customer service manager, sales administrator, help desk agent, help desk manager, and help desk administrator job roles. Analyses and the underlying data is secured through a set of delivered OTBI transaction analysis duty roles. These duty roles are assigned to the service representatives, service managers, help desk agents, and help desk managers, and determine what analyses can be accessed by each. These OTBI transaction analysis duty roles control the subject areas and analyses a user can access. These roles also control the data that the signed in user can see in the analyses. This aligns with the data security privileges for the user in the transaction system.

The administrator defines which users, application roles, and catalog groups have the following privileges:

- Receive the delivery content of an agent.
- Have permission to access a section or alert section in a dashboard.
- Have permission to use a saved modification.
- Have permission to add or edit for an existing catalog group.
- Assign permissions to a catalog object.

The OTBI transactional analysis duty roles for Engagement Cloud are described in detail in the topic "About Security Roles: Explained". The following table lists the Service analyses, and the job and duty role mapping required for a user to access them.

<table>
<thead>
<tr>
<th>Infolet Name</th>
<th>Job Role</th>
<th>OTBI Transactional Analysis Duty Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Agents with Most Pending Service Requests</td>
<td>• Customer Service Manager</td>
<td>• Service Managerial Transaction Analysis Duty</td>
</tr>
<tr>
<td>• Long Wait on Agent</td>
<td>• Help Desk Manager</td>
<td>• HR Help Desk Manager Transaction Analysis Duty</td>
</tr>
<tr>
<td></td>
<td>• Sales Administrator</td>
<td></td>
</tr>
</tbody>
</table>
Creating analyses begins with subject areas. A subject area is a functional grouping of metrics (also called facts) and the contextual entities (also called dimensions) they can be analyzed by. Subject areas are the building blocks of analytic content in OTBI. For example, the columns in a tabular report showing the number of open SRs (metric or fact) by Agent Name (context or dimension) are sourced from one of the subject areas. Multiple subject areas can be joined to produce reports, when cross functional analysis is needed.

The following table lists the available Service subject areas. A brief description is also provided to enable you to build dynamic analysis using one or many of these subject areas at a time. Additional subject areas focused on analyzing knowledge articles and their usage in service requests are available for knowledge managers and analysts. For more information, see Related Topics.

<table>
<thead>
<tr>
<th>Subject Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service - CRM Customer Coverage Real Time</td>
<td>This subject area provides the ability to report on the detail of service request coverage. Working from date ranges, the Service - CRM Customer Coverage Real Time subject area is suitable for cross subject area queries to provide customer level detail. This is a real time subject area, allowing up to the moment reporting on Customer Coverage.</td>
</tr>
<tr>
<td>Subject Area</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Service - CRM Inbound Messages Real Time</td>
<td>This subject area allows users to develop reporting that tracks inbound communications by message channel, and the resulting service requests from those messages. Metric supporting averages for time to create an SR and time to acknowledge message to the seconds, allow a granular understanding of inbound message flow and the resulting SR response and conclusion. This is a real time subject area, allowing up to the moment reporting on Inbound Messages.</td>
</tr>
<tr>
<td>Service - CRM Service Queue Resources Real Time</td>
<td>This subject area allows reporting and analyses around queues, teams, and resources within a service organization. This subject area can be combined with any other subject area that has employee and service queue dimensions, when the results are needed at the employee and service queue levels. For cross subject area queries, include at least one metric from each subject area, and ensure that the session variable ENABLE_DIMENSIONALITY = 1 is set. This is a real time subject area, allowing up to the moment reporting on service queues resources.</td>
</tr>
<tr>
<td>Service - CRM Service Request Messages Real Time</td>
<td>This subject area provides the ability to develop analyses around messages created within an SR. Access to the message channel, associated SR, message details, and message-specific metrics supports the development of complex reporting around SR messages within a real time context. This is a real time subject area, allowing up to the moment reporting on SR messages.</td>
</tr>
<tr>
<td>Service - CRM Service Request Milestones Real Time</td>
<td>This subject area provides the ability to report on the performance of each service agent, and how to resolve SRs and meet default targets. A milestone is a specific target set for the resolution of the SR. The information provided in the CRM Service Request Milestones subject area allows service managers to identify efficient agents, coach, and measure under-performance within the team. It also allows agents to effectively monitor personal performance to meet the milestone targets. This is a real time subject area, allowing up to the moment reporting on SR performance to milestone targets.</td>
</tr>
<tr>
<td>Service - CRM Service Request Resource Real Time</td>
<td>Used to analyze and report on all the resources, and resource roles on an SR. Employee dimension in this subject area refers to both primary and nonprimary resources, unlike other subject areas where employee refers to only primary resource. This subject area can be combined with any other subject areas that have SR dimensions when results are required at the level of an SR. For cross subject area queries, include at least one metric from each subject area.</td>
</tr>
<tr>
<td>Service - CRM Service Requests Real Time</td>
<td>This subject area provides the ability to report on service requests. The information provided in the CRM Service Request subject area allows service managers and agents to review critical SRs, targets, priority status and measure performance within the team or as an individual. It also allows agents to effectively monitor personal performance to meet Service Request targets. This is a real time subject area, allowing up to the moment reporting on Service Requests.</td>
</tr>
<tr>
<td>Service - CRM Social Post Real Time</td>
<td>This subject area provides the ability to develop analyses around social posts created around an SR. Access to the social post channel, associated SR, social post detail, and social post specific metrics supports the development of complex reporting around SR social posts within a real time context. This is a real time subject area, allowing up to the moment reporting on social posts.</td>
</tr>
<tr>
<td>Service - CRM Service Request Action Plan Action Real Time</td>
<td>Enables analysis that helps customer service managers to obtain a complete picture of action plan enforcement of company policies and procedures. Agents can monitor their own performance regarding action plan completions and delays. Action plan information captures compliance against a predefined set of actions, and analytics highlights metrics around open and closed action plans, delayed actions, and aging of action plans. Real-time reporting built using this subject area provides both summary and detail metrics.</td>
</tr>
<tr>
<td>CRM - CRM Interactions Real Time</td>
<td>Interaction information is captured whenever an inbound or outbound communication occurs between service personnel and a customer. This subject area enables you to build analyses to obtain a real time view of the nature of interactions such as the number and frequency of interactions, whether they’re inbound, outbound, and so on, across the spectrum of customers, contacts, channels, and agents. Performance indicators such as average interaction handling</td>
</tr>
<tr>
<td>Subject Area</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CRM - CRM Interaction Service Requests Real Time</td>
<td>This subject area helps you to analyze interactions in the context of SRs and helps answer questions such as the average number of interactions it takes to close an SR, how would that vary between regions, customer segments, products, service categories, and such other business contexts. Such insights help managers identify outliers to take corrective action.</td>
</tr>
<tr>
<td>CRM - CRM Interaction Aggregate</td>
<td>This subject area can be used independently or in conjunction with CRM - CRM Service Request Summary, to help obtain a summarized view of SR interactions. Managers can obtain an insight into the volume of interactions that occur with or without an associated SR, how often these occur, and across what channels.</td>
</tr>
<tr>
<td>CRM - CRM Service Request Summary</td>
<td>Analyses built using this subject area enable customer service managers to obtain a complete picture of SR performance, as it relates to interactions, compliance, and knowledge article usage. The analyses built serve to expand the real-time reporting of service interactions by providing additional summary metrics and prebuilt reports. The added analytic capabilities provide valuable insight, helping service organizations assess the volume and effectiveness of their channel interactions in resolving service issues. Similarly, SR compliance analysis delivers a comprehensive view of SR milestone attainments and compliance. Customer service managers and executives can gain visibility into the diverse factors affecting compliance, identify patterns for a better understanding of this key area of service performance, and take corrective action. Analyses built using this subject area also help managers to track the extent to which agents use knowledge articles in SRs for issue resolutions.</td>
</tr>
<tr>
<td>CRM - CRM Omnichannel Events Real Time</td>
<td>Enables users to build dynamic analyses to monitor Omnichannel work assignments across agent, queue, and channel. Managers can balance agents’ work load by obtaining insight into their current work load, presence, and availability. Key performance indicators such as average customer wait times, duration in queues, offers, and agents’ acceptance rates of work assignments can be measured, for better throughput, higher application efficiency, and improvement in customer satisfaction.</td>
</tr>
<tr>
<td>CRM - CRM Work Orders Real Time</td>
<td>The analysis built using this subject area provides the ability for managers to keep close tabs on SR generated field work orders to ensure alignment with customer commitments and SLAs governing the underlying SRs. Agents can monitor the progress of work orders tied to their own SRs. Key dates such as resolution due date and scheduled date are compared with actual completion dates across resources, teams, products, and customers. This enables you to spot trends and exceptions, and take corrective action or plan better, to avoid future recurrence of sub-optimal performance.</td>
</tr>
</tbody>
</table>

**Related Topics**

- Analyzing Knowledge

## Saved Queries for Service Analytics

Saved queries are canned analyses that enable you to construct reports efficiently. The saved queries are pre-calculated data filters that can be conveniently applied on any analyses that you want to build. They’re also useful in cross subject area reporting.

To access the saved queries do the following:

1. Click **Navigator > Tools > Reports and Analytics.**
2. Click **Browse Catalog**.

The **Oracle Business Intelligence Catalog** page is displayed.

3. In the Folders pane, click **Shared Folders > Service > Subject Area Contents > Saved Queries - OTBI**.

The following table shows the available saved queries.

<table>
<thead>
<tr>
<th>Analyses Name</th>
<th>Description</th>
<th>Job Role</th>
</tr>
</thead>
</table>
| SRs assigned to me                     | Saved query to retrieve the list of SRs assigned to the signed in user. Apply this on the **Service Request ID** column when used in analyses. | • Customer Service Representative  
• HR Help Desk Agent 
• Sales Administrator |
| SRs where I am on the team             | Saved query to retrieve the list of SRs where the signed in user is part of the SR team. Apply this on the **Service Request ID** column when used in analyses. | • Customer Service Representative  
• HR Help Desk Agent 
• Sales Administrator |
| SRs assigned to me or my subordinates  | Saved query to retrieve the list of SRs assigned to the signed in user or the user's subordinates. Apply this on the **Service Request ID** column when used in analyses. | • Customer Service Manager  
• HR Help Desk Manager 
• Sales Administrator |
| SRs where I am or my subordinates are on the team | Saved query to retrieve the list of SRs where the signed in user or the user's subordinates are part of the SR team. Apply this on the **Service Request ID** column when used in analyses. | • Customer Service Manager  
• HR Help Desk Manager 
• Sales Administrator |
| Queues where I am a resource           | Saved query to retrieve the list of queues where the signed in user or the user's subordinates are resources. Apply this on the **Queue ID** column when used in analyses. | • Customer Service Representative  
• HR Help Desk Agent 
• Sales Administrator |
| Queues where I am or my subordinates are resources | Saved query to retrieve the list of queues where the signed in user or the user's subordinates are resources. Apply this on the **Queue ID** column when used in analyses. | • Customer Service Manager  
• HR Help Desk Manager 
• Sales Administrator |
| Work Orders assigned to me             | Saved query to retrieve the list of work orders assigned to the signed in user. Apply this as a filter on the **Work Order ID** column when used in analyses. | • Customer Service Representative  
• Sales Administrator |
| Work Orders assigned to me or my subordinates | Saved query to retrieve the list of work orders assigned to the signed in user or the user's subordinates. Apply this as a filter on the **Work Order ID** column when used in analyses. | • Customer Service Manager  
• Sales Administrator |
Extending Oracle Cloud Applications: Explained

Extensibility is a feature that enables you to modify Oracle Cloud applications by making changes to the associated elements. You can configure applications by adding or modifying associated elements, such as objects, fields, workflow, and security privileges. In Oracle Engagement Cloud, you can configure channels, service requests, messages, queues, and other objects. To learn more about Extensibility, see Oracle Sales Cloud Configuring Sales guide.

Modifying the Service Request Pages

You can modify the pages for Engagement Cloud by duplicating the standard layout and updating the changes you want. The following procedure describes how to modify the SR page layouts.

1. Sign in to Engagement Cloud as an administrator.
2. From the Settings and Actions menu, select Manage Sandboxes. You can either create a new sandbox, or use an existing one and set it to active.
3. After you activate the sandbox, from the Navigation panel, go to Application Composer.
4. In the Application Composer, select Service from the Applications drop-down list.
5. In the Standard Objects list, select Service Request > Pages.

The Service Request: Pages page contains sections for the different layouts available for modification. You can either use the standard layout provided with Engagement Cloud, or duplicate the standard layout and modify the following:

- **Landing Page Layouts**: The landing page layout contains sections to modify the SR summary table, the buttons and actions, and the fields for performing a mass update on the page. Click the edit icon to add, remove, or reorder the fields in each of these sections.
- **Create Page Layouts**: The create page layout contains sections for the Create Service Request page that the users see when they create a new SR. The fields that you define in this layout determine the information that the users are asked to enter when creating the SR. Click the Edit icon to add, remove, and reorder the fields and the buttons available on this form.
- **Details Page Layout**: The Details page layout is the screen that users see when they open an SR to view or edit details. On this layout, you can add, edit, and remove fields from the Summary, Contacts, Team, Linked Articles, and Milestone details tabs, that are displayed as part of the Edit Service Request page. Click each subtab and select the edit icon to make changes to the fields. Not all tabs that are available are extensible. You can only rename the tabs that are not extensible.
- **Service Request Spotlight Region**: The spotlight region is the summary of an SR that is displayed when the SR is opened for viewing or editing details. You can add, remove, and reorder the fields that you want to display on this header. You can also add the current milestone details to the spotlight region to highlight the information.
- **HR Help Desk Service Request**: This layout displays to the HR help desk agents and managers. You can add, remove, and reorder the fields that you want to display for the summary table, buttons, and actions.

6. After you make the required updates, save the layout.
7. To use this layout to view the SRs, select Active to make the layout active for users.
You can also configure the service request severity color codes to display on the Service Requests Lists page. For more information about setting color codes for severity, see “Adding Service Request Severity Values: Procedure.”

**Related Topics**
- Adding Service Request Severity Values: Procedure
- Additional Sales Cloud Configurations and Integrations: Overview

## Extending the Service Request Layout in Mobile: Explained

You can extend your mobile application to modify the fields that display as part of your service layout. You can also modify the layout based on specific user privileges.

By default, the Standard Layout is enabled for the Service Request feature in the mobile application. You can duplicate this standard view to create a new layout, and edit the following aspects:

- Add, remove, and rearrange the fields that you want to display.
- Select and assign roles to the layout. If you select specific roles, only users with those roles can view the layout in their mobile application. For example, display partner-specific details only for partner users.
- Select and assign regions to the layout.
- Select and assign advanced criteria. This enables you to define a set of conditions that have to be met before the page layout is displayed for a feature's Detail or Edit views.

Do the following to modify your service layout for Oracle CX Cloud Mobile using Application Composer.

1. Sign in to Engagement Cloud as an administrator.
2. From the **Settings and Actions** menu, select **Manage Sandboxes**. You can either create a new sandbox, or use an existing one and set it to active.
3. After you activate the sandbox, from the **Navigator** menu, click **Application Composer**.
4. Ensure that **CRM Cloud** is selected in the **Application** drop-down list.
5. Under the **Common Setup** list of options, select **Mobile Application Setup**.
6. Click **CX Cloud Mobile**. The CX Cloud Mobile section is displayed.
7. Select **Service Requests** in the **Application Features** list.
8. Select **Duplicate** against the Standard Layout to create a new layout.
9. Enter a name for the new layout and click **OK**.
10. From the **Layout Details** page, select the fields, roles, regions, and advanced criteria that you want to assign to the layouts. The following are the available layouts and their descriptions:
   - List: The list view displays a list of SRs with some basic details. You can modify this view to display up to 12 fields for each SR.
   - Detail: The detail view displays the drilled down details for a specific SR. This view displays when the user selects an SR from the list to view details.
     - To enable the **Business Unit** field in the SR details view, drag the **Business Unit Name** field to the layout.
   - Edit: The edit view contains the fields that a user sees when creating or editing an SR. Ensure that you enable all fields that you require in this view.
     - To enable the **Business Unit** drop-down list in the Create and Edit SR pages, drag the **BUOrgId** field to the layout.
Saved Search: The saved search view displays the search options that you want to provide to the users. When you set a saved search as default, the SRs in the list view display the results from the default saved search. These saved searches display on SRs list view for the users.

You can rearrange or remove the attributes on the mobile view by clicking and dragging the attribute to your preferred location. The mobile view also simulates how the display looks on the application screen for the users.

11. Click **Save**.
12. Download the sandbox on the client and test before you publish the changes.

**Related Topics**
- Configuring Oracle CX Cloud Mobile: Explained
- Testing Oracle CX Cloud Mobile Configurations: Worked Example
- Oracle CX Cloud Mobile: Overview

### Setting Up and Defining Groovy Notification Triggers: Explained

A notification is an alert for service agents and managers, to notify them about an event on the service request, and enable them to take the necessary actions.

When the **Service Notifications** feature is enabled, bell notifications are automatically enabled for service requests. To enable push notifications on mobile devices, an administrator must enable the **Mobile Notifications** feature under **Service Notifications**. An administrator can define notification triggers as Groovy scripts in **Application Composer**, containing the conditions that must be met for each notification. Notifications are triggered when the defined conditions are met. For example, you can define notification triggers for upcoming milestones, reassignments, or escalations.

**Note:** Notifications can be temporarily disabled using the profile option SVC_DISABLE_BO_NOTIFICATIONS.

### Enabling Notifications for Service Requests

To enable the notifications feature for service requests, do the following:

1. Navigate to **Setup and Maintenance**.
2. Select the **Service** offering.
3. In the **Functional Areas** section, select **Productivity Tools > Change Feature Selection**. The Edit Features: Productivity Tools page is displayed.
4. Select **Service Notifications**.
5. To enable push notifications for mobile devices, select **Mobile Notifications**.

### Defining Notification Triggers

To define a Groovy script for a notification trigger, do the following:

1. Click your user image or name in the global header, and select **Manage Sandboxes** under the **Administration** menu. The **Manage Sandboxes** window is displayed.
2. Create a sandbox or select an existing one, and click **Set as Active** to activate the sandbox. The sandbox is designated as the active sandbox.
3. Close the **Manage Sandboxes** window.
4. In the **Navigator** menu, select **Application Composer**. The Application Composer page is displayed.
5. Under **Objects**, select **Standard Objects > Service Request > Server Scripts**. The **Server Scripts Service Request** section is displayed.
6. Click the **Triggers** tab.
7. From the **Action** menu, click **Add**. The **Create Object Trigger** section is displayed.
8. Create a Groovy trigger:
   a. Specify the **Name**, **Error Message**, and **Trigger Definition** details.
   
   **Note:** For information about the trigger type and APIs to use in your Groovy script, see the table in the following section "Sample Groovy Scripts".
   b. Click **Save and Close**.

**Debugging Groovy Triggers**
Within your Groovy scripts, all your `println` statements go to the runtime messages. To view the runtime log and debug your Groovy script, do the following:

1. Within a sandbox, navigate to **Application Composer**.
2. Under **Common Setup**, select **Run Time Messages**. The **Run Time Messages** section is displayed.
3. Select the **Enable Application Script Logging** check box.
4. Trigger your notifications.

If the Groovy script has appropriate `println` statements, you receive runtime messages when you navigate back to the **Run Time Messages** section in Application Composer.

**Sample Groovy Scripts**
This section provides sample Groovy scripts you can use as a reference to trigger notifications.

The following table provides guidance on the trigger types and APIs to use within Groovy triggers.

<table>
<thead>
<tr>
<th>Trigger Types</th>
<th>Available APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is generally recommended to use the trigger type <strong>After Changes Posted to Database</strong>. This enables you to stop potential issues if the Groovy script is accidentally written to run infinitely. If the trigger type is set to <strong>Before Update to Database</strong>, with a bad script, there may be some issues.</td>
<td>When you create a Groovy script, you need the API names of the fields you’re trying to access. To create your triggers based on parent/child fields, do the following:</td>
</tr>
</tbody>
</table>
| **Note:** The `isAttributeChanged()` function works only for the **Before** trigger types. The workaround for the **After** trigger types involves retrieving the old value before the database is updated, then retrieving the new value after the | 1. In the **Trigger Definition** section, click **Show/Hide Expression Palette**.  
2. Click the **Fields** tab.  
3. Select an **Object**.  
4. Click the **Maximize Edit Script** arrow. The fields for the selected object are displayed.  
5. Select the API you want, and click **Insert**.  
6. To close the expression palette, click the **Restore Edit Script** arrow, then click **Show/Hide Expression Palette**. |
If you're creating a new object, and you want to trigger on creation, it is recommended to use the trigger type **Before Insert to Database**. However, some of the **Before** trigger types do not have all attributes exposed yet. This results in some fields being blank. To debug your triggers if you're not getting the expected results, follow the steps in the previous section "Debugging Groovy Triggers".

Use the following code to send a notification to the service representative when the SR is assigned (for a new or existing SR).

```java
// Send notification to assignee when SR is assigned
// Set groovy trigger type to "After Changes Posted to Database"
int indexOfAssignee = adf.oldValue.getAttributeIndexOf('AssigneePartyId')
def oldValue = adf.oldValue.getAttribute(indexOfAssignee,oracle.jbo.server.EntityImpl.TRANS_ORIGINAL_VERSION);
def newValue = getAttribute('AssigneePartyId');

// Check whether the value in the database has been changed. This avoids cases where you save the SR without changing the status.
// If you do not compare oldValue to newValue, a notification will always be sent in this case.
if ((oldValue != newValue)) {
    try {
        def messageText = 'This SR has been assigned to you - ' + SrNumber + ':' + Title
        def recipientPartyId = AssigneeResourceId
        if (recipientPartyId) {
            adf.util.sendNotification(adf, messageText, recipientPartyId)
        // For debugging purpose, you can use println statements as in the following example. This will print to the Run Time Messages log.
        //println("Notification sent to " + AssigneePersonName + " because SR is assigned")
        } catch (e) {
            throw new oracle.jbo.ValidationException('Failure: ' + e.getMessage())
        }
    }
}
```

Use the following code to send a notification to the primary contact when the SR is resolved.

```java
// Send notification to primary contact when SR is resolved
// Set groovy trigger type to "After ChangesPosted to Database"
int indexOfStatus = adf.oldValue.getAttributeIndexOf('StatusCd')
def oldValue = adf.oldValue.getAttribute(indexOfStatus,oracle.jbo.server.EntityImpl.TRANS_ORIGINAL_VERSION);
def newValue = getAttribute('StatusCd');
```
// Check whether the value in the database has been changed. This avoids cases where you save the SR without changing the status.
// If you do not compare oldValue to newValue, a notification will always be sent in this case.
if ((oldValue != newValue) && StatusCd == 'ORA_SVC_RESOLVED') {
    try {
        def messageText = 'This SR has been resolved - ' + SrNumber + ': ' + Title
        def recipientPartyId = PrimaryContactPartyId

        // If there is no recipientPartyId, in this case, if there is no PrimaryContact, no notification will be sent.
        if (recipientPartyId) {
            adf.util.sendNotification(adf, messageText, recipientPartyId)
            // For debugging purpose, you can use println statements as in the following example. This will print to the Run Time Messages log.
            // println("Notification sent to " + PrimaryContactPartyName + " because status of SR is Resolved")
        }
    } catch (e) {
        throw new oracle.jbo.ValidationException('Failure: ' + e.getMessage())
    }
}

Use the following code to send a notification to the service representative when the SR is escalated:

//Send notification to assignee when SR's Critical Flag is raised.
//Fire only if CriticalFlag is set.
//Set groovy trigger type to "Before Update to Database".
if (isAttributeChanged('CriticalFlag') && CriticalFlag=='Y') {
    try {
        //For debugging purpose, you can uncomment the following lines and check Run Time Messages to see the logs.
        //def logLine = 'Trigger.SrEscalation:Raising SR Escalation notification : ' + 'Sr Number-Sr Title: ' + SrNumber + ' - ' + Title
        //println(logLine)
        def messageText = SrNumber + ' - ' + Title + ' is now critical!
        def recipientPartyId = AssigneePartyId

        adf.util.sendNotification(adf, messageText, recipientPartyId)
    } catch (e) {
        throw new oracle.jbo.ValidationException('Failure: ' + e.getMessage())
    }
}

Setting up Service Requests for Partners: Explained

Service Requests (SR) can be used to capture and resolve issues reported by your partner accounts. Instead of associating customer accounts and customer contacts to the SR, you can associate partner accounts and partner contacts. Partner users or partner contacts can use the simplified user interface to submit service requests and to view any responses returned by an internal resource that owns the SR.

Note: The service offering provides only one global service product catalog and one global category hierarchy. All service requests, whether they’re associated with customer accounts or partner accounts, display the same hierarchy of products and categories. Partners can only create and interchange an SR with partners through the simplified user interface, the SR APIs, and the inbound and outbound emails. The CTI framework and the chat channel are not available for partners.

To enable creating service requests specific to partners, you must modify the page layouts for the service request object. The layout must be modified according to the privileges provided to the external partner contact and the internal resources.
that access the SR. For detailed information about modifying layouts using the application composer, see Extending the Applications for Functional Administrators.

The users that can access the partners SR can be broadly classified as internal and external resources. External resources are partner contacts that create, submit and view an SR for the partner that they represent. Internal resources own and work on the SRs for one or more partners that they’re assigned to. Oracle recommends that you provide access to partner service requests using the following standard roles. However, you can grant access to other standard and user-defined roles that you create, depending on your requirements.

- Partner sales representative - An employee from a partner company that can request support on issues. In the Partner Management work area, these users are also represented as partner contacts.
- Channel account manager (CAM) - Internal resource responsible for a set of partner accounts. You can grant access to this role if you want them to respond to the partner service requests in addition to their other duties.
- CRM administrator - Configures the application, such as assignment rules, service product catalog, queues, layouts, extensibility changes, and so on.

When a partner account is associated with the SR, the Primary Contact field is always treated as a partner contact. The primary contact field enables users to search and select only from the partner contacts associated with the partner account that is selected. You can also modify the SR list view for the partner contact roles, to include columns specific to the partner service requests.

Modifying the Service Request Page Layout for External Resources

External resource in partner contacts must be provided limited access to creating and editing a partner SR. The following list describes the recommended fields that you must enable and disable in the SR layout for external resources.

- Remove internal only fields such as Queue, Assigned To.
- Add Groovy to make the Partner Account, Status and Channel Type read-only.
- Remove the Attachment + icon. Partner users can only upload and view attachments through a specific SR message.
- Enable only the Summary and Message subtabs.
- (Optional) Allow the user to set the severity when creating an SR. However, the user cannot change this when editing the SR.
- Enable the Milestones tab.

In addition to modifying the SR layout page, remove the Service Request action in the Contacts page layout using the application composer.

Modifying the Service Request Page Layout for Internal Resources

Internal resources are resources that work on the partner SR. The following list describes the recommended fields that you must add to the SR layout for internal resources.

- Add Partner Account field to the SR page layout. Since the primary contacts are chosen based on the partner account selected, display the Partner Account field first, before the Primary Contact field.
- (Optional) Remove the Account field from the layout to avoid confusing the user. For partner service requests, the Account field is only for reference information.

In addition to creating page layouts for the partner SR work area, you can also expose the Service Request sub tab on the Partner 360 work area.
Assigning Partner Service Requests to Queues

If you have a different set of resources responding to customer and partner service requests, you can set up separate queues to address the partner SRs.

To assign a partner SR to a queue, do the following.

1. Navigate to **Setup and Maintenance**.
2. Click the **Setup** drop-down list and select **Service**.
3. Select the **Service Request** functional area, and then select **Manage Service Request Assignment Objects** task.
4. In the **Manage Service Request Assignment Objects** page, select **Service Request** and the select **Attributes**.
5. In the **Attributes** tab, add the **Partner Account** field and save the changes.
6. In Setup and Maintenance home, search and select the task **Manage Service Request Assignment Rules**.
7. Add rules to ensure that when the **Partner Account** field is populated, the service request must be assigned to a different set of queues. For more information about assigning service requests to queues, see Define Service Request Assignment Rules.
17 Importing and Exporting

File-Based Data Import and Export: Overview

You can import and export a wide range of application data using file-based data import and export in Oracle Engagement Cloud. For example, you can use the file-based data export feature to export object data so that you can then import it into another instance. You can also import records to the applications so that you don’t have to create the records in the user interface. Only users with the service request administrator duty role can import and export objects. For more information, see the Oracle Sales Cloud - Understanding File-Based Data Import and Export guide.

Importing Objects

You can import several objects into Engagement Cloud using the file-based import.

To create a new file import activity, sign in to Setup and Maintenance as an administrator, and search and select the task Manage File Import Activities. For high volume data import, select the ODI-based import when importing the service requests. The ODI-based import is only available for Service Request object and its child objects. When you select the Service Request object, ensure that you don’t select the Execute Groovy scripts and workflows option in the Import Options section. Selecting the Execute Groovy scripts and workflows option executes an ADF-based import which can be slow for large volume import activities.

Before you import service requests, queues, or categories, you must first import the following dependent objects:

- Accounts: The list of accounts that can be associated with the SR. Selecting an account is required when creating an SR. For more information about importing accounts, see Importing Accounts in the Oracle Sales Cloud Understanding File-Based Data Import and Export guide.

- Contacts: The list of contacts for each account that can be associated with the SR. For information about importing contacts, see Importing Contacts in the Oracle Sales Cloud Understanding File-Based Data Import and Export guide.

- Employee Resources: The list of employee resources to which you can assign work objects in the SR. For information about importing employee resources, see Importing Employee Resources in the Oracle Sales Cloud Understanding File-Based Data Import and Export guide.

- Partners: The list of partner accounts that you can associate with an SR. For information about importing partners, see Importing Partners in the Oracle Sales Cloud Understanding File-Based Data Import and Export guide.

- Product Groups: A product group is a group of related products. For information about importing product groups, see Importing Product Groups in the Oracle Sales Cloud Understanding File-Based Data Import and Export guide.

- Products: List of products against which you can raise an SR. For information about importing products, see Importing Products in the Oracle Sales Cloud Understanding File-Based Data Import and Export guide.

- Asset: List of assets associated with the account.

- Categories: The categories associated with the service requests.

- Queues: The queues to which your SRs are assigned.

- Channels: Import the channels that are already associated with your service requests.
The following list describes the top-level and sublevel objects that you can import into the service application. For more information about these objects, refer the Service Requests section in the File-Based Data Import for Oracle Sales Cloud guide.

- Service Requests
  - Messages
  - Contact Members
  - Resource Members
  - References
- Queues
  - Queue Party Resources
  - Queue Team Resources
- Categories
  - Service categories

Note:
- Since the service request data is dependent upon queues and categories, you must import the queues and categories before importing your service requests.

- Interactions: You cannot import interactions into Engagement Cloud, but you can export the interactions for the service requests.
- Inbound email
- Inbound Message Filters: You can import the inbound message filters through Setup and Maintenance.
- Milestone Configuration: You can import and export milestone configurations only through Setup and Maintenance.
- Channels
  - Channel Resources
- Standard Text Folders: The import facility doesn’t support importing hierarchical data directly. To import folders hierarchies, such as parent and child relationships, you must import the standard text folders data twice. The first time you import the file, the object data is added, and the second time you import the same file, the relationships are created.
- Standard Text Variables
- Standard Text
  - Standard Text Relations
- Self-Service Roles

After you import the dependent objects, create an import activity to upload details of objects you want to upload. Importing an object involves completing the following steps:

1. Set up the import options.
   You select the object you’re importing, specify the file format and different import options, and upload the file with your data. The options that are available depend on the object that you’re importing. Some import objects, for example, permit you to upload a ZIP or Java archive (JAR) file of attachments. These attachments are then attached to the records in the application after the data file import is complete.
2. Map the fields.

For this task, you create a mapping of the data in your file and the attributes in the application. You can also reuse a mapping that you have created in past import activities or use the Manage File Import Mappings task. Any mapping you create in an activity is automatically saved. This mapping can be reused in subsequent imports and can be managed using the Manage File Import Mappings task.

3. Schedule the import.

You can schedule to run the import at a time you specify or run it immediately.

4. Review and activate the import activity.

Review your import details and click **Activate** to activate the import activity.

You can use predefined .csv templates for your service requests, queues, and categories that are provided in the application. You can also import user-defined objects.

**Exporting objects**

You can extract large volumes of data from the Oracle Engagement Cloud using bulk export. You can either extract a full set of records for an object, or perform incremental extracts. For example, you can extract complete set of service requests data or extract updated set of records every week.

*Related Topics*

- Using Predefined Templates to Import Data Through File-Based Data Import
- File-Based Data Import and Export of Custom Objects: Explained
- File-Based Data Import for Oracle Sales Cloud

**Bulk Export: Overview**

You can extract large volumes of data from Oracle Sales Cloud objects using bulk export. You can either extract a full set of records for an object, or perform incremental extracts. For example, you can extract complete set of account data or extract updated set of records every week. Bulk export creates comma separated or tab delimited files, which are attached to the export process.
The following figure depicts the process of selecting data for export, scheduling the export activity, and delivering the data file to the customer application.
18 Integrating Oracle Field Service Cloud

Oracle Field Service Cloud: Overview

Oracle Engagement Cloud and Oracle Field Service Cloud integration is designed to support customers who want to synchronize Engagement Cloud work orders with activities in Field Service Cloud. With the integration, service work orders can be synchronized with activities in Oracle Field Service Cloud. The synchronization is executed using Oracle Integration Cloud Service as the backbone for mapping and information exchange.

The integration supports the following:

- Creation of an Oracle Engagement Cloud service work order which triggers creation of an Oracle Field Service Cloud activity.
- Creation of an Oracle Field Service Cloud activity from an Oracle Engagement Cloud service work order including:
  - Update
  - Reschedule
  - Cancel

- Updating Oracle Engagement Cloud service work orders from Oracle Field Service Cloud including:
  - Update
  - Move
  - Start
  - Suspend
  - Cancel
  - Not Done
  - Complete

For more information about integrating Oracle Engagement Cloud with Oracle Field Service Cloud, see the Oracle Engagement Cloud Integrating Oracle Engagement Cloud with Oracle Field Service Cloud guide on Oracle Help Center at https://docs.oracle.com.

Related Topics
- Engagement Cloud Integrating Oracle Engagement Cloud with Oracle Field Service Cloud
Integration Component Architecture Between Oracle Engagement Cloud and Oracle Field Service Cloud: Explained

Service work order management is the primary use case handled in the Oracle Engagement Cloud and Oracle Field Service Cloud integration.

Service work order management consists of both work order creation and updates in Oracle Engagement Cloud and updates in Oracle Field Service Cloud. To achieve this integration, a combination of point-to-point and bidirectional integrations are used. The point-to-point integrations are used for retrieving data from Oracle Field Service Cloud used in the creation and rescheduling of work orders. Bidirectional integration is used for synchronizing Oracle Engagement Cloud work orders with Oracle Field Service Cloud activities. Oracle Engagement Cloud and Oracle Field Service Cloud Bi-Directional integration use Oracle Integration Cloud Services as the integration component. Integration Cloud Services is a complete, secure, and lightweight integration solution where you can connect your applications in the cloud. It simplifies connectivity between your applications, and can connect both your applications that exist in the cloud and your applications that are still maintained on-premise.

The integration manages error handling and guaranteed delivery by introducing concrete fault handling and prevention measures in the integration layer that are realized through Integration Cloud Services (ICS). The integration domain covers typical elements and integration functionality, such as adapters for connectivity to back-end systems, routing, transformation, and filtering.

The following figure shows the process flow of information between Engagement Cloud, Integration Cloud Services or Oracle Integration Cloud, and Field Service Cloud.

The following figures show the point-to-point components of the Oracle Engagement Cloud and Oracle Field Service Cloud integration using the Oracle Field Service Cloud Capacity API to retrieve the data work order area list in Oracle Engagement Cloud from Oracle Field Service Cloud.
Oracle Engagement Cloud Integration Services

The Oracle Engagement Cloud web services `CustomerWorkOrderService` is used in the integration. This SOAP API is called from the Event Handling Framework to retrieve a work order and Integration Cloud Services to create, update, reschedule, and cancel a work order in Oracle Engagement Cloud.

Oracle Field Service Cloud Integration Services

The following Oracle Field Service Cloud web services are used in the integration:

- **BulkUpdateActivity** REST API. Use this web service through ICS to create, update, and reschedule an activity in Oracle Field Service Cloud.
- **CancelActivity** REST API. Use this web service through ICS to cancel an activity in Oracle Field Service Cloud.
- **Capacity** SOAP API. Use this web service in the point-to-point integration when creating and scheduling a work order to retrieve the list of work order areas based on postal code and time zone and the scheduler data based on work order area and work order type.

Oracle Integration Cloud Services

The prebuilt integrations are available through Oracle Marketplace. You can sign in and install the package directly into your Integration Cloud Service (ICS) instance. The installation includes the following:

- Connection: Engagement Cloud R13
- Connection: Field Service Cloud R13
- Integration: OEC OFSC Work Order Created R13
- Integration: OEC OFSC Work Order Updated R13
- Integration: OEC OFSC Work Order Canceled R13
- Integration: OFSC OEC Activity Updated R13
Note: If Oracle Marketplace is not available, you can download the prebuilt files from My Oracle Support. To access the prebuilt integration flow, see Integrating Oracle Engagement Cloud with Oracle Field Service Cloud on My Oracle Support. Oracle Support Document 2247612.1 In the Attachments section, select the For Release 13 implementations attachment. Save the OEC_OFSC.par file to a local computer.

Overview of Echo Suppression and Bi-Directional Synchronization

During bidirectional Oracle Engagement Cloud work order and Oracle Field Service Cloud activity synchronization echoes are generated. This means that when an event is triggered in Oracle Engagement Cloud it is synchronized through Integration Cloud Services to Oracle Field Service Cloud, which then fires an event in Oracle Field Service Cloud and then back to Oracle Engagement Cloud, on and on. The Integration Cloud Services-based integration uses an echo suppression mechanism, which stops unwanted update or create events (the echoes) from going back to the source application.

Caution: You must follow the user name guidelines for the Oracle Engagement Cloud integration and Oracle Field Service Cloud integration because they’re used for echo suppression in the prebuilt integration flows. If you use different user names, you must modify the prebuilt integration flows in ICS for echo suppression to work.
19 Integrating Oracle Internet of Things Service Monitoring for Connected Assets

Oracle Internet of Things Service Monitoring for Connected Assets: Overview

Oracle Engagement Cloud and Oracle Internet of Things Service Monitoring for Connected Assets integration is designed to support customers who want to take advantage of the latest capabilities of the Engagement Cloud application to service customers, while using Internet of Things Service Monitoring for Connected Assets to monitor connected assets.

With the integration, assets connected to the internet periodically report information back to the Internet of Things Service Monitoring for Connected Assets application. This data is constantly monitored to detect problems indicating a failure has occurred or is imminent, and reports the issue to Engagement Cloud for customer service follow-up. The synchronization is executed using Oracle Integration Cloud as the backbone for mapping and information exchange.

The integration supports the following:

- Creation of an Internet of Things Service Monitoring for Connected Assets incident when business rules detect that a failure has occurred or is imminent, which triggers creation of an Engagement Cloud service request.
- Closing the Internet of Things Service Monitoring for Connected Assets incident when the corresponding Engagement Cloud service request is resolved.
- Viewing information about the connected asset directly from the service request pages in Engagement Cloud. This enables the service agent to retrieve both historical and the most current diagnostic information from the device, and take remote actions against the device (for example, restart), for troubleshooting and issue resolution.

For more information about Internet of Things Service Monitoring for Connected Assets, see https://docs.oracle.com/en/cloud/saas/iot-asset-cloud/osmca-start/index.html.

Integrating Engagement Cloud with Internet of Things Service Monitoring for Connected Assets: Explained

This topic describes how to integrate Oracle Engagement Cloud with Oracle Internet of Things Service Monitoring for Connected Assets.

To integrate Engagement Cloud with Internet of Things Service Monitoring for Connected Assets, do the following:

1. Create an integration user with privileges to perform the integration with Internet of Things Service Monitoring for Connected Assets.
2. Import the integration package from Oracle Marketplace.
3. Import SSL certificates for Engagement Cloud and Internet of Things Service Monitoring for Connected Assets, if the instances use self-signed certificates.
4. Activate connections to Engagement Cloud and Internet of Things Service Monitoring for Connected Assets.
5. Activate the integrations.
6. Enable the Connected Asset tab.
7. Import the Internet of Things connected assets.

For more information about configuration and integration steps for Internet of Things Service Monitoring for Connected Assets, see https://docs.oracle.com/en/cloud/saas/iot-asset-cloud/osmca-start/index.html.

Create an Integration User for Internet of Things Service Monitoring for Connected Assets

To integrate Engagement Cloud with Internet of Things Service Monitoring for Connected Assets, Oracle recommends that you create a user specifically for the integration. The integration user can call the Sales Cloud service catalog or event catalog web services from Oracle Integration Cloud.

The following procedure describes how to create an integration user and what privileges to provide to the role.

1. Sign in to Oracle Sales Cloud as an administrator.
2. Select Navigator > My Team > Users, Roles and Delegations, and click Manage Users.
3. On the Manage Users page, select Create New User from the Actions menu.
4. Enter the following details for the new user:
   - Last Name: Enter a last name for the user.
   - e-mail: Enter a valid email ID for the user.
   - Hire Date: Select today’s date.
   - User Name: Enter a user name for the user.
   - Person Type: Select Employee from the drop-down list.
   - Legal Employer: Select the legal employer from the list.
   - Business Unit: Select a valid business unit.
   - Send user name and password: Select this option.
   - User Log in: Enter the user name you created.
   - Password: Enter the password for the user.
5. Save the user details. An email is sent to the address after the user has been created.
6. Check the user credentials sent in the email, sign in as the new user and reset the password. After creating the user, sign in to the security console and provide the following roles to the integration user.
   - SOA Operator
     The SOA Operator is a duty role and duty role cannot be directly assigned to a login using security console. Create an enterprise role as a parent of SOA Operator and associate that enterprise role to the integration user.
   - Customer Service Representative
   - Resource

Import the Integration Package

After you have set up Engagement Cloud and created the integration user with the required privileges, you can set up the Integration Cloud integration package.

Before you start setting up Integration Cloud, go to the Oracle Marketplace and search and download the Engagement Cloud to Internet of Things Service Monitoring for Connected Assets integration package. After you download this package, perform 
the following procedures to set up Integration Cloud. For more information about Oracle Marketplace, see the Oracle Cloud Marketplace documentation.

The Integration Cloud integration package supports the following:

- Inbound: Create an SR in Engagement Cloud for an Internet of Things Service Monitoring for Connected Assets incident.
- Outbound: Respond to Internet of Things Service Monitoring for Connected Assets on the SR status.

Import SSL Certificates

If the instances use self-signed certificates, you must import the SSL certificates before you configure and activate the connections.


Activate the Connections to Engagement Cloud and Internet of Things Service Monitoring for Connected Assets

After you download the package and import the SSL certificates, connect to Engagement Cloud using the Sales Cloud adapter. The procedure for importing the adapter is described in the topic "Creating an Oracle Sales Cloud Adapter Connection", available at https://docs.oracle.com/en/cloud/paas/integration-cloud-service/icssc/index.html.

The following procedure describes how to configure the connections to the Sales Cloud instance.

1. Sign in to the Integration Cloud application using your integration user credentials.
2. Click the Connections icon on the home page.
3. Ensure that Oracle Engagement Cloud is listed on the connections page.
4. Click Oracle Engagement Cloud to view the details.
5. Click Configure Connectivity and enter the following information in the Connection Properties dialog box:
   - OSC Service Catalog WSDL URL: Enter the service catalog URL on your Engagement Cloud instance.
   - (Optional) OSC Event Catalog URL: Enter the event catalog URL on your Engagement Cloud instance.
6. Click OK.
7. Click Configure Security and enter the following information:
   - Security Policy: Enter Username Password Token.
   - User name: Enter the integration user name.
   - Password: Enter the password for the integration user.
   - Confirm Password: Reenter the password.
8. Click OK.
9. Click Test on the Actions bar of the Sales Cloud page.
10. Ensure the connection test is successful and the status meter shows 100%.
11. Click OK.
The following procedure describes how to configure the connection to the Internet of Things Service Monitoring for Connected Assets instance.

1. Sign in to the Integration Cloud application using your integration user credentials.
2. Select Connections and then create a new connection for Internet of Things Service Monitoring for Connected Assets.
3. Select Rest Adapter Connection.
4. Click Configure Connectivity and enter the following information:
   a. Connection Type: REST API base URL
   b. Connection URL: Internet of Things Service Monitoring for Connected Assets instance base URL
6. Save the configuration and test the connection.

Activate the Integrations
You must activate the following integrations.

- Autocreate SR for IoT Incident: Creates or updates an SR in Engagement Cloud from an Internet of Things Service Monitoring for Connected Assets incident by mapping the object attributes of the two applications.
- Push SR Status to IoT: Updates the Internet of Things Service Monitoring for Connected Assets incident with the SR status from Engagement Cloud.

After you configure the connections, activate the integrations as follows:

1. Sign in to the Integration Cloud application using your integration user credentials.
2. Click the Integrations icon on the home page.
3. Search for each integration by its name.
4. Click the Activate button on each integration.
5. Select the Enable detailed tracing option, and click Activate on the confirmation dialog box.
6. Ensure the flow has been activated successfully.
7. Repeat the previous steps for each integration.

Enable the Connected Asset Tab
This section describes how to enable the Connected Asset tab on the user interface.

To enable the Connected Asset tab, do the following:

1. Import the SSL certificate from the Internet of Things Service Monitoring for Connected Assets instance to the Engagement Cloud server.
2. Sign in to Engagement Cloud as a setup or administrator user.
3. On the Setup and Maintenance page, search for the Manage Service to IoT Cloud Integration task.
4. Enter the Internet of Things Service Monitoring for Connected Assets instance URL, user name, and password.
5. Click Verify Connection to verify the connection.
6. Save the configuration details.

Import Internet of Things Connected Assets
To import Internet of Things connected assets from an external data source, see “Importing Assets” in the Oracle Sales Cloud Understanding File-Based Data Import and Export guide, at http://www.oracle.com/pls/topic/lookup?ctx=cloud&id=OAFDI1745520.
20 Implementing Digital Customer Service

Overview of Digital Customer Service

Oracle Digital Customer Service is an offering within Oracle Engagement Cloud that enables you to provide your customer account users self-service access to their service requests and relevant knowledge articles through a web interface.

The Digital Customer Service application user interface can be administrator-defined to reflect a company brand. Using Oracle Visual Builder Cloud Service, you apply themes and templates, and include various UI components, depending on your business needs.

Once configured and deployed, your customer account’s users can self-register as Digital Customer Service users. Once the self-registration is complete, Digital Customer Service users can sign in to the Digital Customer Service application UI to communicate with your customer service representatives using a web interface, chat or co-browse.

The following figure provides a general overview of the Digital Customer Service architecture.


About Digital Customer Service Terminology

This topic describes terminology related to the Digital Customer Service offering in Oracle Engagement Cloud. Some terminology is related to other software components.

- **Oracle Visual Builder Cloud Service**: A visual development tool for creating applications by dragging and dropping UI components onto a page. You can create business objects and add data to your application. The tools and platform on which the Digital Customer Service components are built are integrated with Oracle Engagement
Cloud and the associated APIs. Refer to the related topics that follow for more information about terminology specific to Oracle Visual Builder Cloud Service.

- **Theme**: HTML and Cascading Style Sheet (CSS) files that provide the look and feel of the implementing company’s brand. The theme enables the implementing company to style the application to match branding experience.

- **UI Components**: The items that you place onto a web page that provide key capabilities within a user experience, for example, knowledge search, and results.

- **Business Objects**: The definitions of data objects that enable UI components to interact with Oracle Engagement Cloud APIs.


- **Digital Customer Service Users**: Your customer account users who have successfully self-registered to use the Digital Customer Service application. These users can have a variety of roles.

- **Digital Customer Service Templates**: The available templates you can select while creating your Digital Customer Service application. These application templates include component extensions, themes, and depending on the template, predefined pages and actions.

- **Digital Customer Service Reference Implementation**: This template includes several pages and business components that enable basic support experience including: product browsing, knowledge search, service request creation and management, chat, and self-service user management capabilities for the account administrator. The administrator manages all of the users and roles.

- **Digital Customer Service Starter**: This template is an empty application that makes the Digital Customer Service components available in the component palette of Oracle Visual Builder Cloud Service.

**Related Topics**

- **Getting Started with Oracle Visual Builder Cloud Service**

- **About Digital Customer Service Roles**
Glossary

**lookup code**
An option available within a lookup type, such as the lookup code BLUE within the lookup type COLORS.

**lookup type**
The label for a static list that has lookup codes as its values.

**profile option**
User preferences and system configuration options that users can configure to control application behavior at different levels of an enterprise.

**report**
An output of select data in a predefined format that’s optimized for printing.

**resource**
People designated as able to be assigned to work objects, for example, service agents, sales managers, or partner contacts. A sales manager and partner contact can be assigned to work on a lead or opportunity. A service agent can be assigned to a service request.

**resource organization**
An organization whose members are resources. Resource organizations are used to implement sales organizations, partner organizations, and so on.

**resource role**
Resource roles indicate the role a resource plays as an individual, or within a resource team.

**resource team**
A resource team is a temporary group of resources formed to work on work objects. A resource team may contain a resource organization or resources or both. A resource team cannot be hierarchically structured and is not intended to implement an organization.