

Oracle Fusion Service

**Implementing Service Center with
the Redwood User Experience**



Oracle Fusion Service
Implementing Service Center with the Redwood User Experience

F77885-26

Copyright © 2024, Oracle and/or its affiliates.

Author: Oracle Fusion Service Information Development Team

Contents

Get Help i

1 About This Guide 1

Audience and Scope	1
Related Guides	1

2 Access Service Center 3

How do I enable Service Center?	3
What are the ready-to-use privileges and roles?	4

3 Profile Options, Lookups, and Scheduled Processes for Service Requests 7

What are Profile Options, Lookups, and Scheduled Processes?	7
Overview of Profile Options	7
Lookup Types	8
How do I change lookups?	9
Modify Channel Type Visibility	10
FAQs for Lookups and Lookup Types	11
What are scheduled processes?	12
Profile Options and Scheduled Processes for SR Management	13
Purge SRs and Related Objects When You Purge Contacts	21
Audit Trail for Service Requests	22
FAQs About SR Management	23

4 Users and Security 27

Overview of Defining Setup Users	27
Overview of Setting Up Users and Security	27
What are resources?	29
Setting Up Resource Teams	30
FAQs for Define Resource Team Information	32
About Security Roles	33
Set Up Service Request Visibility Based on Queue	35

Set Up Service Request Visibility Based on BU	36
Data Security Policies on SRs	38
Set Up Read-Only Access to the Service Request Header	43
Set Up Read-Only Access to the Service Request Header and Service Request Messages	44
5 Set Up Activities Configurations	47
What Service Request Activities Track and Display	47
What are the Activities setup tasks?	48
Required Job Roles and Privileges for Creating Activities Configurations	49
How You Enable and Display Activities	50
How do I create and activate a service request activities configuration?	51
Before You Run Activities Initialization Jobs	52
Specify What Displays in Activities	53
How do I run the Publish Feed Elements job to index activities?	55
How do I migrate an index of activities from production to test?	57
6 Extend Service Center	59
How do I extend my Service Center application?	59
Extensibility at a Glance	59
Set Up Visual Builder Studio	61
How to Provision a VB Studio Project Using an Application Template	61
Service Request Standard Pages Extensibility Matrix	61
Extend Service Center navigation	63
Disable the Service Center dashboard button	65
Modify any field on posted service request messages in Redwood	65
Work with Dynamic UI	65
Work with Dynamic Forms	68
Work with Dynamic Tables	78
Work with Dynamic Containers	79
Service Request Categories	82
UI Events Framework	83
7 Catalog Management	151
How do I define a catalog for Service?	151
Overview of Sales Catalogs	152
Sales Catalog High-Level Setup Steps	153

Add Products to the Catalog	155
Validate the Service Catalog	156

8 Configure Service Request Outcomes and Resolutions 157

Outcomes and Resolutions	157
Manage Outcomes and Resolutions	157
Ready to Use Outcomes and Resolutions	158
Manage Resolutions	159

9 Work Assignment and Routing 161

Overview of Assignments, Mappings, and Rules	161
Implementation Concepts for Work Assignment	162
Assignment Mappings	163
Assignment Rules	164
Tips for Creating Assignment Rules and Rule Sets	169
Work Assignments	172
Omnichannel	178
Prevent Work Starvation	181
FAQs About Service Requests and Queues	183

10 Service Entitlements 187

Overview of Service Entitlements	187
Milestones and Coverages	188
Set Up Additional Condition Columns for Standard Coverage Entitlement Rules	198
Set Up Administrator-Defined Milestones	201
Configure Milestones for Multiple Business Units	204
Troubleshoot with Milestone Diagnostics	206

11 Set Up Service Business Units 207

How are business units for set up for Service and Help Desk?	207
How do I enable multiple Business Units Fusion Service?	208
Manage Internal Resource Organizations for Service BUs	209
Manage Resource Organization Hierarchies for Service BUs	209
Create a Business Unit for Service	210
Associate Resource Organizations with Business Units	211
Create Employees for a Service BU	211

How do I set the scope in Service BU setup?	213
Overview of optional setup tasks for Service BU	214
Secure the Business Unit Field on a Service Request	215
Export and Import the Functional Setup Data for Business Units	216
Overview of BU-Based Service Request Visibility	218
Migrate Business Objects Between Different Business Units	218
Associate Different Catalogs with Different Business Units	219

12 Import and Export 221

What's Import and Export Management?	221
Use Functional Setup Manager to Import, Export, and Compare Categories Between Environments	222
Before You Import	223
Import Objects	223
Overview of Bulk Data Export	224

13 Real-Time Channels 227

Chat	227
Computer Telephony Integration (CTI)	240
CTI Media Toolbars APIs	245
Configuration APIs	251
Interaction APIs	254
Event Listener APIs	260
Configure Screen Pop Pages	269
Call Flow Parameter Settings	283
Configure Wrap Up	284

14 Set Up Digital Assistant as a Chat Bot 295

How do I create Oracle Digital Assistant as an agent?	295
Create a User Agent for Digital Assistant	296
Configure Chat Options and Enable Notifications	296
Set Up Queues and Agent Capacity for the Bot User	297
Extend the Universal Work Object	298
How do I create routing rules?	298
How do I synchronize users and roles to Oracle Identity Cloud Service?	300
Link with Oracle Identity Cloud Service	301
Create a Resource Application	301

Configure a Client Application	302
Provision Chat Service	302
Configure Oracle Digital Assistant	302
Create and Extend an ODA Skills	303
Create a New Authentication Service	304
Set Up Digital Assistant as an Agent in Oracle Digital Assistant	305

15 Email Channels 307

How are inbound messages processed?	307
How do I set up inbound and outbound email?	309
Hide the message identification number in response emails	310
Enable Configuration of Email Communications	310
How do I configure an email channel?	311
Inbound Message Filters	313
How do I configure profile options for inbound and outbound email?	316
How do I enable the ability to add all contacts and team members or add additional email recipients?	321
How do I set the From Name in outbound emails?	321
How do I define email templates?	322
How do I change acknowledgment messages for inbound email?	323
Decide Whether to Add Email Recipients as Service Request Contacts or Team Members	324
How do I enable deep links to KM articles in outbound emails?	324
Access Point Setup for Inbound Emails	327
Job to Process Inbound Emails	335
Display Email Thread in Preview and Outbound Email	336
Purge Inbound Email Messages	341
Send Bulk Emails	343
Detect Unprocessed Emails	344
FAQs for Service Email	345
Notes	348

16 Integrate with Microsoft Exchange Server 353

Overview of the Integration with Microsoft Exchange Server	353
Register an App in Microsoft Azure	353
Limit Access of Mailboxes Using REST API	356
Configure Fusion Email	357
How do I create a job to read messages?	359
Create a Job to Delete Messages	360

How do I schedule deletion of inbound emails that were previously retrieved from Exchange Server?	361

17 Interactions 363

Interactions	363
Configure Cross-Channel Interactions	364
Data Security Policies for Interactions	366

18 Action Plans 369

Set Up Action Plans	369
Use Oracle Process Automation with Action Plans	381
Configure Oracle Process Automation with Oracle Fusion Service	383

19 Integrate with Intelligent Advisor 385

Oracle Intelligent Advisor	385
Create a Fusion Sales and Fusion Service Connection in Intelligent Advisor	386
Download and Install Policy Modeling	386
Connect the Policy Modeling Project to Intelligent Advisor Hub	386
Map Data to and from Fusion Sales and Fusion Service to Policy Modeling	387
Configure Intelligent Advisor Interviews	387

20 Set Up Agent Insights 389

Overview of the Insights Panel	389
Set Up the Agent Insights Panel	391

21 Enable Productivity Tools 393

How do I enable productivity tools?	393
How do I manage SmartText entries?	394
How do I create and edit private SmartText entries?	395
How do I create and edit public SmartText entries?	396
What are the SmartText naming conventions?	397

22 Channels for Collaboration 399

How do I control the display of channels used only for internal conversations?	399
How do I create an email channel for collaboration or internal conversations?	400
Translate the Communication Channel Display Names	400

23	Slack Channels	401
	Overview	401
	Set Up a Slack Channel: Method 1	402
	Set Up a Slack Channel: Method 2	406
	Slack Notifications	413
	User Credentials for Executing Commands	416
	FAQs About Slack Integration	416
24	Collaborate with Slack	419
	How do I collaborate with Slack?	419
	Create a Collaboration Action for Slack	419
	Create a Child Action for a Collaboration Action for Slack	422
	Create a Second-Level Child Action for a Collaboration Action for Slack	423
	Add an Event to Capture Messages Mentioning Your Slack App	424
25	Collaborate with Microsoft Teams	425
	Prerequisites for Setting Up a Microsoft Teams Channel for Fusion Service	425
	How do I Set Up a Microsoft Teams Channel?	425
	Create a Microsoft Teams App	426
	Set Up a Bot User for Your Microsoft Teams App	427
	Register Your Microsoft Teams App in the Azure Portal	427
	Enable Collaboration With Microsoft Teams in Functional Setup Manager	428
	Update Functional Setup Manager with Credentials of Your Microsoft Teams App	428
	Create a Microsoft Teams Channel	429
	Send and Receive Attachments from Microsoft Teams Channel Users	429
	Enable Notifications to Microsoft Teams	430
	Create a Collaboration Action	430
	Create a Child Action for a Collaboration Action	432
	Create a Child Action for the Reply to Request Child Action	433
	Configure the FND_IDP_PROXY_USER_WHITELIST Profile Option	435
	Upload Your Microsoft Teams App	435
	How do I install the Microsoft Teams app and add it to my team?	436
26	Configure Notifications	437
	Set Notification Triggers and Preferences	437
	Sample Groovy Scripts for Notifications	446

Set Push Notifications	452

27 Adaptive Intelligence **453**

Overview of Adaptive Intelligence	453
Use attachment attributes in Adaptive Search of service requests	453
Configure Service Request Similarity	453
Configure Service Request Classification	455

28 Set Up Service Work Orders **469**

Enable Work Orders	469
Extend Work Orders UI Elements	469

29 Set Up Survey Integration **471**

Can I use surveys?	471
Survey Requirements and Licensing	471
Use the Survey Framework with Oracle Integration	472
Create the Fusion Service Administrator User for Connecting to Oracle Integration	472
Download and Save the Oracle Integration Mapping Package	472
Configure the Integration Connections	473
Configure the Decision Model	478
Activate the Survey Integration Flow	479
Note Your Integration Endpoints	479
Create Your Survey	480
Supported Survey Question Types	481
Enable Surveys in Fusion Service	481
Manage Survey Products	482
Manage Survey Objects	482
Manage Survey Configurations	483
Manage Survey Templates Task	484
Schedule Jobs for Sending Surveys and Collecting Survey Responses	485
Verify the Survey Integration is Working	485
How to Include an Administrator-Defined Attribute in the Decision Model	486
Export and Import the Functional Setup Data for Surveys	489
How to Change an Active Decision Model	490
What are the prerequisites for survey integration using SurveyMonkey?	491
FAQs About Survey Integration	492

30	Service Analytics	497
	How do I view analyses using OTBI?	497
	Service Analytics Roles	497
	Analytics Subject Areas	499
	Overview of Parent and Child Interactions in Analytics	499
	Service Analytics Saved Queries	499
	Where do I find resources for Oracle Fusion Service Analytics?	500
31	Set Up Additional Service Features	503
	Installed Base Assets	503
	Case Management	504
	Service Request Parts Ordering for Service Oracle Service Logistics Cloud Users	505
	Copy Maps	506
32	Integrate Oracle Internet of Things Service Monitoring for Connected Assets	509
	Overview of Oracle Internet of Things Service Monitoring for Connected Assets	509
	Integrate Engagement Cloud with Internet of Things Service Monitoring for Connected Assets	509

Get Help

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

Get Help in the Applications

Some application pages have help icons  to give you access to contextual help. 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. If the page has contextual help, help icons will appear.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at [Oracle University](#).

Join Our Community

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

Share Your Feedback

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

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

Thanks for helping us improve our user assistance!

1 About This Guide

Audience and Scope

If you're looking for information about getting up and running with your Service Center application, then this guide is for you. The information is primarily aimed at service administrators and managers in your service organization.

This guide describes how to:

- Enable the Redwood user interface in Functional Setup Manager
- Grant the required privilege to the roles of your users
- Use Visual Builder Studio to extend the layout of the user interface

Related Guides

To understand more about the information covered in this guide, refer to the list of guides in the following table.

Guide Names	Description
<i>Using Service Center in the Redwood User Experience</i>	Contains information to help service personnel and other service end users to perform day-to-day business tasks using Service Center in the Redwood User Experience.
<i>Extending Oracle Fusion Applications with Visual Builder Studio</i>	Describes how to use Visual Builder Studio to extend your application.
<i>Using Fusion Service</i>	Contains information to help service managers, service personnel, and other service end users to perform day-to-day business tasks using Oracle Fusion Service.
<i>Implementing Fusion Service</i>	Describes how to set up Service Request Management.
<i>Securing Fusion Sales and Service</i>	Contains information to help setup users and sales administrators configure access to functionality and data.
<i>Security Reference for Fusion Sales and Fusion Service</i>	Lists the predefined security data that's included in the offering.

2 Access Service Center

How do I enable Service Center?

To enable Service Center functionality and UI, you just need to opt-in from Fusion Service.

Note: Once you opt in to the Redwood User Interface for Service feature to enable the Redwood UI make sure to then click the pencil icon to ensure that the Service Center sub features are also enabled. There are two sub features to review are Service Requests for Redwood User Interface and Work Orders for Redwood User Interface. These separate sub features allow the administrator to control whether each set of object-related pages should be enabled in Redwood.

Here are the steps:

1. Sign in to the application as an administrator or setup user.
2. Click **Navigator > My Enterprise > Offerings**.
3. On the Offerings page, select the **Service** offering.
4. Click the **Opt In Features** button.
5. On the **Opt In: Service** page, review the functional areas, such as Service Requests, Channel Communication and so on.
6. Scroll through the list until you find **Redwood User Interface for Service**.
7. Select **Enable**.
8. Click the **Features** (pencil) icon.
9. On the Edit Features: Redwood User Interface for Service page, enable all child features.
10. Click **Done**, and then on the Opt In: Service page, click **Done** again.

Your users can now access Service Center including Work Orders (Next Generation).

Disable the Service Center homepage if needed

In certain case, you might want to disable the Service Center homepage. Do do so, you admin must do the following:

1. In Setup and Maintenance, click  the search and select **Manage Profile Options**.
2. On the Manage Profile Options page, click  to add a new record.
3. Create a new profile option using the following table:

Column	Values
Profile Option Code	SVC_ENABLE_REDWOOD_SERVICE_HOME
Profile Display Name	Enable Redwood Service Home Page
Application	Service
Module	Service

Column	Values
Description	Enable Redwood Service homepage.
SQL Validation	<code>Select meaning lookup_code from fusion.fnd_lockups where lookup_type='YES_NO'</code>
Start Date	Enter the current date.

4. In the Profile Option Levels area, enable the **Site** and the **User** levels then save your work.

What are the ready-to-use privileges and roles?

The following privileges are ready-to-use with the following default job roles: Customer Service Representative and Customer Service Manager:

Note: The View Service Center privilege only grants users the ability to access each of the Redwood Service Center pages. Within each page, the more detailed functionality is controlled by existing functional privileges, most of which apply to both the Visual Builder and ADF UI pages concurrently. Any custom roles you may have created also require you to add the SVC_USE_SERVICE_BASE_APP privilege to allow that role access to Service Center.

The following table shows the ready-to-use privileges.

Note: In case you don't find the privileges in Security Console, run the **Import User and Role Application Security Data** process from **Navigator > Tools > Scheduled Processes**.

Privilege	Privilege Code
View Service Center	SVC_VIEW_SERVICE_CENTER_PRIV
View Navigator Menu	ORA_ZCA_VIEW_NAVIGATOR_MENU_PRIV
View Smart Actions	ZCA_VIEW_SMART_ACTIONS_PRIV
View Named Routes	ORA_ZCA_GET_NAMED_ROUTES_PRIV
Manage Installed Base Assets	SVC_MANAGE_INSTALLED_BASE_ASSETS_PRIV
Open Objects in Dynamic Tabs	SVC_OPEN_OBJECTS_IN_DTNT

Note: If you're using custom roles, it's recommended that you add the following duty role as a child of your custom role: Conversation Message Management (ORA_SVC_CONV_MESSAGE_MANAGEMENT).

The privileges shown in the table enable users to access a new link in the Home Page and Navigator within the Service grouping called Service Center. If you've previously created your own roles, you must use Security Console to add these privileges to your roles.

The following table shows two more privileges required for Service Center for the Redwood User Experience. You must create a custom role or use an existing custom role and manually add these two privileges to the custom role.

Privilege	Privilege Code
Use Service Base App	SVC_USE_SERVICE_BASE_APP
Navigate to Service Center Pages from Notification	SVC_NAVIGATE_TO_SERVICE_CENTER_FROM_NOTIFICATIONS_PRIV

To expose the **Cases** feature, use Security Console to provide the following roles to your users:

- Case Manager: ORA_SVC_CASE_MANAGER
- Case Worker: ORA_SVC_CASE_WORKER

3 Profile Options, Lookups, and Scheduled Processes for Service Requests

What are Profile Options, Lookups, and Scheduled Processes?

Profile options, lookup types, and scheduled processes let you configure application behavior and process data.

Briefly, here's what profile options, lookup types, and scheduled processes do:

- 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: Act on data in the applications.

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

Related Topics

- [What are scheduled processes?](#)
- [How can I access predefined profile options?](#)
- [How can I access predefined lookups?](#)

Overview of Profile Options

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. This table lists some examples:

Type of Information	Profile Option Setting Example
User preferences	Set preferences at the user level
Installation information	Identify the location of a portal
Configuration choices	Change UI skins and actions
Processing options	Determine how much information to log

Type of Information	Profile Option Setting Example
Enabling the Activity Stream	Enable agents to see the Activity Stream

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 levels that are allowed to be set are preconfigured with the application.

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

1. **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's the highest level.
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. **User:** This level affects only the current user. It has the highest precedence, over Site and Product.

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.

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

Level	Priority	Example
Site	Lowest	Currency for a site is set to Euros.
Product	Supersedes Site	Currency for the product or set of products is set to UK pound sterling.
User	Highest, supersedes Product	Currency for a user is set to US dollars.

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

Related Topics

- [Set Profile Option Values](#)
- [How can I access predefined profile options?](#)

Lookup Types

Lookup types provide the lists of values in application fields that are drop-down lists.

Note: You can't create a new lookup with the name starting with ORA, to avoid any conflict with the lookups predefined in the application.

How You Modify Lookup Types

You can change 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.

Here's a table that shows which lookup management tasks are allowed at each modification level.

Allowed Task	User	Extensible	System
Deleting a lookup type	Yes	No	No
Inserting new codes	Yes	Yes	No
Changing the wording that displays on the page (Meaning field)	Yes	Yes	Yes
Updating start date, end date, and enabled fields	Yes	Yes, only if the code isn't predefined data	No
Deleting codes	Yes	Yes, only if the code isn't predefined data	No
Updating tags	Yes	No	No
Updating module	Yes	No	No

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 changed. The configuration level for lookup types created using the Define Lookups page is by default set at the User level.

How do I change lookups?

Administrators can change 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
- Waiting
- Resolved
- 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 as "In Progress - Troubleshooting" versus "In Progress - Repairing." This procedure maps one or more Statuses to status types.

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

- Manage Service Request Internal Priorities
- Manage Service Request Sources
- Manage Service Request Resolutions
- Manage Service Request Outcomes and Resolutions
- Manage Service Request Severities
- Manage Service Request Channel Types
- Manage Service Request Channel Type Visibility
- Manage Service Request Problem Types
- Manage Service Request Contact Relationship Types
- Manage Service Request Status Values
- Manage Service Request Profile Options
- Manage Service Request Categories
- Manage Service Request Knowledge Profile Options
- Manage Service Request Parts Order Disposition Options

To change lookups:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Service Request
 - Task: Select all tasks and click the task you want to change.
2. In **Lookup Codes**, click the lookup code that you want to change.
3. Change the fields to correspond to your needs.
4. Click **Save and Close**.
5. Repeat the procedure for all the service request lookup tasks.

The service request lookup is changed.

Modify Channel Type Visibility

Customer administrators can disable channels that should not be displayed in the application when an agent composes a message of type customer entry or response.

1. Sign in to Fusion application as a setup user or administrator.
2. In the Setup and Maintenance work area, go to the **Tasks** panel tab.
3. Search for and click the **Manage Service Request Channel Type Visibility** task.
4. On the Manage Service Request Channel Type Visibility page, if you don't see the channels you want to disable, click Sync on the toolbar of the table.
5. For any channel that you select, you see two rows with different values for **Property Name**:
 - **Enabled for Collaboration**: To enable the selected channel to collaborate with subject matter experts to resolve customer issues, set the **Property Value** for this property name to **Yes**.

- **Enabled for Customer Interaction:** To enable using the selected channel for interaction of your service agents with your customers, set the **Property Value** for this property name to **Yes**.

Note: You can also use the drop-down list in the toolbar for the **Property Name** field to filter the list by each property name.

6. Set the **Property Value** to **No** to disable the selected channel.

7. **Note:** For messages of type **Customer Entry** and **Response**:

- SMS and Slack channels are disabled by default because they aren't supported for customer communications. You can enable or disable the other channels.
- Although the Social channel is disabled by default, Social channels are visible to agents if the SR is created for a social post using the social channel integration. For other SRs, the Social channels are disabled. Oracle recommends that you keep this channel disabled.

8. Click **Save and Close**.

The disabled channels aren't displayed when an agent composes a message.

FAQs for Lookups and Lookup Types

Frequently Asked Questions (FAQs) for Lookups and Lookup Types

This section contains some frequently asked questions (FAQs) when setting up and maintaining Lookups and Lookup Types. Scroll or search to find answers to common questions.

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. You can edit lookups using the following tasks in the Setup and Maintenance work area:

- 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 central to an application. However, lookup types defined for a specific application are managed using the task for that application.

Can I create a new lookup name starting with ORA?

No, you can't create a new lookup with the name starting with ORA. The application validates the lookup names to avoid any conflict with the lookups seeded in the application.

What are scheduled processes?

Scheduled processes do tasks that are too complex or time-consuming to do manually, for example importing data or updating many records. You can run scheduled processes on a recurring schedule and send notifications based on how the process ends.

Some scheduled processes give you printable output. Those processes might have **Report** in their name.

Use the Scheduled Processes work area to run all the processes you've access to and to manage submissions. If you need access to this work area, ask your security administrator to assign you a role that gives you access, for example a custom role with the the Manage Scheduled Processes (FND_MANAGE_SCHEDULED_PROCESSES_PRIV) privilege. Other than the Scheduled Processes work area, you might be also able to run certain processes from other work areas.

Jobs and Job Definitions

Each scheduled process that you run is based on a job. The job is the executable that controls what the process does. Each job needs a job definition, which, for example, identifies the parameters and other options you've set for the process. You or your administrator can create job definitions for Oracle Analytics Publisher reports so that users can run the reports as scheduled processes.

Process Sets

A process set is a scheduled process that contains multiple processes or even other process sets. So, when you submit a process set, you're running more than process.

Note: When you submit certain scheduled processes, the job logic causes other processes to automatically run. But in this case, you're not submitting a process set that includes those other processes.

Submission

When you submit a scheduled process, you can use its parameters to control how, and which records are processed. For example, a process includes only the transactions that were edited by the person you select for a Last Updated By parameter. Some scheduled processes don't have parameters.

As part of the submission, you can also set up a schedule for the process, for example to run once a week for two months. Every time a process runs, there's a unique process ID.

Output

Some scheduled processes provide output in PDF, HTML, and other formats. For example, a process can import records and also produce output with details about those records. There are many types of output, for example a tax document or a list of transactions.

Profile Options and Scheduled Processes for SR Management

What are the SR profile options?

Service request profile options let you configure and control application data. Administrators and setup users can manage SR profile options in the Setup and Maintenance work area.

You can set various profile options and schedule job processes for service requests. Some 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 describes the various profile options for service request management. For more information on scheduled processes, see [What are scheduled processes?](#)

Profile Option	Description
SVC_ASSIGN_TO_QUEUE_ON_CREATE	Assigns an SR to a queue automatically, when the SR is created. Note: The preferred way of assigning an SR to a queue is by using Omnichannel assignment. If Omnichannel is enabled, this profile option is ignored.
SVC_SR_IN_RESOLVED_DAYS	Defines the number of days after which any resolved SR is automatically closed. This profile option must be used along with the <i>Auto-Close Service Requests</i> job process. 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.

Profile Option	Description
	The SVC_SR_IN_RESOLVED_DAYS profile value must be set to 0 to disable the Auto-Close Service Request job.
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 <i>Auto-Close Service Requests</i> job process. 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 <i>Auto-Close Service Requests</i> job process.
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 <i>Auto-Close Service Requests</i> job process.
SVC_PUID_FORMAT	Defines the format for the unique reference number on each SR. Every SR number consists of a prefix and radix. Let's say that you select SR_ as the prefix and No alphanumeric conversion as the radix. Without this formatting profile option, the SR numbers are generated as SR_1, SR_2, and so on. Now suppose you use this formatting profile option and the value is 000000. Then the SR numbers are generated as SR_000001, SR_000002, and so on. For more information about configuring the prefix and radix, see How do I configure the prefix and radix? Note: The SVC_PUID_PREFIX profile option used earlier no longer exists.
SVC_SR_DEFAULT_SEVERITY_CD	Sets the default Severity value for a new SR. You can also set this value as blank, if you want the SRs to be created without a Severity value.
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. Although a soft deleted SR is in the database, it can't be retrieved or updated from the UI or REST APIs. Once purged, SRs are completely removed from the database. This profile option must be used along with the <i>Purge Deleted Service Requests</i> job process to purge SRs. When a soft-deleted SR created from an inbound email is purged, all data associated with that SR is also purged from the inbound message tables.
SVC_INBOUND_ENABLE_FAILED_MESSAGE_PURGE	Enables or disables purging of inbound messages that fail or aren't processed successfully. When the value is set to Yes, the purge job deletes the records corresponding to such messages from the inbound message tables.

Profile Option	Description
ORA_SVC_ENABLE_DOWNLOAD_ATTACHMENT	This profile option has two settings. Yes (default) and No. If the profile option is set to Yes, attachments are allowed to be downloaded. To prevent users from being able to download attachments from a Service Request and a Service Request Message, do the following: 1. Set the ORA_SVC_ENABLE_DOWNLOAD_ATTACHMENT profile option value to No. 2. Copy the Service Request Troubleshooter role. 3. Copy the Customer Service Representative role. 4. Edit the custom Service Request Troubleshooter role and delete the Download Attachment of an SR or SR Message privilege. 5. Edit the custom Customer Service Representative role and delete the three Service Request Troubleshooter roles and add the custom Service Request Troubleshooter role. 6. Add the custom Customer Service Representative roles to the users who should not be allowed to download attachments.
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.
SVC_INBOUND_MSG_RETENTION_DAYS	Specifies the number of retention days for successfully processed inbound messages. After the specified number of retention days, the successfully processed inbound messages are deleted from the inbound message tables.
SVC_ENABLE_MESSAGE_CORRECTION	Enables you to edit the text of the Internal Note and Customer Entry SR messages that you've previously created and posted.
SVC_ENABLE_SEVERITY_COLOR_IN_SR	Enables the color code for the service request severity set up in the Manage Service Request Severities task, if set to Yes. The task Manage Service Request Severities enables you to add different severity levels for service requests and define different colors.
ORA_ENABLE_QUEUE_MEMBER_NOTIFICATIONS	Enables selecting all queue members as notification recipients, on the Notification Preferences page. Performance might be impacted if you have many queue members and use this option. Default value is No.
ORA_SVC_DELETE_CLOSED_SR_ENABLED	Enables automatic deletion of an SR that has been closed for at least the number of days specified by the ORA_SVC_CLOSED_SR_RETENTION_DAYS profile option. This profile option must be used along with the Delete Closed Service Requests job process.
ORA_SVC_CLOSED_SR_RETENTION_DAYS	Defines the number of days for which a service request must remain closed before it's deleted. Deletion can't be undone. This action is applicable only when the Delete Closed Service Requests Enabled profile option is set to Yes. This profile option must be used along with the Delete Closed Service Requests job process.

Hide previous customer responses

You can enable the activity stream to display only the most recent customer response to agents. This optimizes the agent's workload because in the activity stream each message is only shown once.

All you need to do is enable a profile option. The profile option isn't enabled by default, so here's how you enable it:

1. In Setup and Maintenance, click  and search and select **Manage Administrator Profile Values**.
2. Enter `ORA_SVC_PREVIOUS_RESPONSE_EMAIL_REMOVAL_ENABLED` in the Profile Option Code field.
3. Set the Site value to **Yes**, then save your changes.

Automatically expand service request messages

Service request messages appear to users in the activity stream where the message content might not all be visible. To view all the message content the user might need to click the Show More link for each message.

To automatically expand all message content, you need to set the profile option `ORA_SVC_MESSAGE_SET_EXPAND_MODE` to Yes. The default value is No.

1. In Setup and Maintenance, click the Tasks icon and search and select **Manage Administrator Profile Values**.
2. Enter `ORA_SVC_MESSAGE_SET_EXPAND_MODE` in the Profile Option Code field.
3. Set the Site value to **Yes**, then save your changes.

How do I configure the prefix and radix?

When a service request (SR) is created, a unique number or ID is generated for it by the application.

Users can't easily read or use these unique IDs because of their length and complexity. As an administrator, you can configure the unique ID that's generated to make it more user-friendly, readable, and specific to your requirement.

This user-friendly value called the public unique ID consists of a prefix and a radix, and you can configure both of them. For example, `SR_0000027413` is a configured public unique ID, where `SR_` is the prefix and `0000027413` is the radix or suffix.

To configure the prefix and radix for SRs:

1. Sign in to the application as a setup user or administrator.
2. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Productivity Tools
 - o Task: Manage Public Unique Identifier Sequence Generation
3. On the Manage Public Unique Identifier Sequence Generation page, for the Service Request **Object Name**, specify a prefix of your choice in the **Prefix** column. For example: `SR-`.
4. Click in the **Radix** column for the Service Request object.

5. From the multiple options displayed in the drop-down list, select the type of radix that you want for your SRs .

This option determines the radix value of the request number. To format the auto generated radix value, you can configure the SVC_PUID_FORMAT profile option as described in the "Profile Options for SR Management" topic.

Note: For more information about the types of public unique IDs, see the Oracle Fusion Cloud Sales Automation Implementing Sales guide.

6. Click **Save and Close**.

Permit SR Contacts that Aren't Related to the Primary SR Account

If you're using the application to serve a customer with an account hierarchy with multiple subsidiaries, it's possible that the primary contact on an SR may be from a different subsidiary than the primary customer account. Or, you may have a third party who has the permissions to report an SR and also interact with the SR. For example, an implementor or partner contact, or a private contractor.

In such scenarios, you would want to remove the built-in validation that requires the SR account and SR contact to be interrelated. And you can permit the following types of primary and secondary contacts on an SR:

- Contacts that belong to multiple accounts
- Contacts with no account
- Contacts associated with a partner account
- All of the above

You can do this by configuring the following profile options.

Profile Option	Description
ORA_SVC_ALLOW_ANY_CONTACT_ON_SR	Specifies whether contacts that aren't related to the SR can be added to an SR. This includes contacts of other accounts, consumer contacts, and partner contacts. The default value is No.
ORA_SVC_DEF_ACCESS_FOR_3RD_PARTY_CONTACTS	Specifies the default access level for third party contacts to view the contents of an SR. The default value is No Access. If you set the value to Regular Access, third party contacts can view an SR. So, contacts who are added to the SR but don't belong to the SR account can view the SR in Digital Customer Service.

What are the Service Request Managements scheduled processes?

You can set various profile options and schedule job processes for service request (SR) management.

You must use some profile options along with the job processes to achieve the results you want. For example, when you set the profile value for closing a resolved SR after N number of days, also 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.

Here's a table that describes the job processes you can schedule to manage SRs. For more information about SR scheduled processes, see the Understanding SR Scheduled Processes guide on Help Center.

Job Process Name	Description
Auto-Close Service Requests	Closes service requests that were resolved some days ago, and resolves service requests that have been in waiting status for some 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. Note: When an SR is closed by this job process, the Status field on the SR can be audited.
Purge Deleted Service Requests	Purges service requests and their child records that were deleted some days ago. The number of days is set using the SVC_SR_IN_DELETED_DAYS profile option.
Purge Inbound Messages	Purges the successfully processed messages from the inbound message database tables, based on the retention days specified in the SVC_INBOUND_MSG_RETENTION_DAYS profile option. If the SVC_INBOUND_ENABLE_FAILED_MESSAGE_PURGE profile option is set to Yes, this job also purges inbound messages that failed or weren't processed successfully.
Load and Update Cloud Metrics for Service	Performs incremental loading and updating of usage and business metrics that are targeted to cloud usage patterns.
Service Request Queue Assignment	Assigns queues to service requests. This job takes the following parameters: - Work Object Code: Indicates business objects that get assigned to agents, such as, service requests. Expected Value: <code>ORA_Service_Request_Work_Object</code> - Candidate Object Code: Indicates objects that are the possible pool of assignment candidates, such as queues. Expected Value: <code>ORA_Queue_Candidate_Object</code> - Assignment Mode (List of Values: Classification, Matching, Scoring, Territory): Indicates the type of assignment processing. Matching is the only mode that's supported. - View Criteria Name: Indicates the view criteria used to identify the service requests to be assigned. Expected Value: <code>OpenSRsUnassignedToQueueByStripeCd</code> - Bind Variable: Indicates the bind variables required for the view criteria. Expected Value: <code>BindStripeCd=ORA_SVC_CRM</code> - 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. - Diagnostic Mode (checkbox): Indicates if the process must be run in diagnostic mode to view the details of assignment processing in an output log.

Job Process Name	Description
	Note: The preferred way of assigning an SR to a queue is by using Omnichannel assignment.
Service Configuration Setup	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.
Aggregate Service Requests	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. Note: If you need faster data refreshes, increase the frequency. However, this impacts the performance of the transaction system.
Monitor Service Request Milestones	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.
Purge Service Event History	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.
Monitor Action Plan Actions	Evaluates the status of an action plan. The recommended frequency for running this job is 15 minutes. Note: If you need faster data refreshes, increase the frequency. However, this impacts the performance of the transaction system.
Execute Incremental Load of Cross-Channel Interaction Data for Reporting	Incrementally loads cross-channel interaction data, for reporting using the CRM - CRM Interaction Aggregate subject area. You must run this job daily to load and update the new data everyday.
Execute Full Load of Cross-Channel Interaction Data for Reporting	Fully loads cross-channel interaction data, or refreshes the data based on a particular date, for reporting using the CRM - CRM Interaction Aggregate subject area. This is an on-demand job. You must run this job in the following scenarios: • The first time data is loaded. • Any time the data needs to be refreshed from a specific date.
Refresh Service Categories for Reporting	Generates the service category hierarchy and stores it in the SVC_CATEGORIES_CF table in a flattened form for ease of reporting. Service category attributes in all service request subject areas obtain data from this table. This process needs to be scheduled to pick up changes to the service category hierarchies to ensure correct reporting. The recommended frequency for running this job is 1 hour.
Migrate Service Business Unit Data	Migrates the following business objects from one business unit (BU) to another: • Category

Job Process Name	Description
	- Channel - Milestone Configuration - Service Request
Execute Incremental Load of SR Audit Data for Reporting	Incrementally loads service request audit data added because the previous run of the process. Use this job to create business intelligence reports using the Service - CRM Service Request Lifecycle subject area. The recommended frequency for running this job is hourly.
Unlock Scheduled Process that Incrementally Loads SR Audit Data	Removes the process lock created by the Execute Incremental Load of SR Audit Data for Reporting scheduled process, when that process fails to remove its lock automatically. This process should be run only if the Execute Incremental Load of SR Audit Data for Reporting scheduled process is unable to start, and no other instance of that process is currently running.
Purge Obsolete User Preferences	Removes the user notification preferences data of invalid users, in case the resources leave your company or are no longer active. This scheduled process lets your organization ensure that your data on user notification preferences isn't stale.
Purge Obsolete Notification Followers	Removes inactive and end-dated users from following the service request, so that the application doesn't get overloaded with stale data. This scheduled process lets you evaluate whether a resource is still valid and enabled to follow specific service requests.
Delete Messages from External Email Applications	This scheduled process deletes inbound email on the Exchange Server.
Delete Closed Service Requests	This scheduled process deletes closed service requests which have been closed for at least the number of days specified by the ORA_SVC_CLOSED_SR_RETENTION_DAYS profile option.

How do I configure a scheduled process?

Here's how you configure a scheduled process:

1. Sign in to the application as an administrator.
2. From the **Navigator**, select **Scheduled Processes**. The Scheduled Processes Overview page is displayed.
3. Click **Schedule New Process**. The **Schedule New Process** dialog box is displayed.
4. 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.

How's a request closed?

A is closed automatically when the scheduled process runs as scheduled.

The auto-close process closes that were resolved a specified number of days ago. The number of days for this scenario is set using the profile .

Related Topics

- [What are the Service Request Managements scheduled processes?](#)
- [What are the SR profile options?](#)

Purge SRs and Related Objects When You Purge Contacts

Decide Which Related Objects to Purge When You Purge Contacts

When a contact is purged, you may have to purge Service objects to comply with General Data Protection Regulations (GDPR).

You can determine whether you want to purge associated service requests (SRs) and related objects in such cases. You can also decide to purge specific objects such as SR messages.

You can decide the actions to take on the SRs and their related objects when contacts are purged from Fusion Sales and Service. To specify the actions, you must configure the **ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT** profile option and select one of the values as required for your company. See [Configure Profile Option for Actions to Take on SRs and Related Objects When Contacts Are Purged](#).

Configure Profile Option for Actions to Take When Contacts Are Purged

Configure the ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT profile option to decide the actions to take on SRs and their related objects when contacts are purged. You can select one of the listed values as required for your company.

All the values for the ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT profile option are described in this table.

Values for the `ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT` profile option

Value of <code>ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT</code> profile option	What it does
Delete SRs and child objects when purging associated contact	Deletes all service requests and their child objects where the purged contact is the primary or secondary contact.
Delete SRs and child objects when purging SR primary contact	Deletes all service requests and their child objects where the purged contact is the primary contact.
Do nothing	Makes no changes to the service request. This is the default behavior of your service application when a contact is deleted.
Remove the association of purged contacts with SR	Removes the association between the purged contacts and the service requests, but make no other changes to the service request or its child objects.
Clear SR Messages with PII data when purged contact is sender	Clears out the content of all the service request messages where the purged contact is the sender and the message is marked to contain Personally Identifiable Information (PII).
Clear SR Messages with PII data where correspondent purged	Clears out the content of all service request messages where the purged contact is a correspondent and the message is marked to contain PII.
Clear SR Messages where the purged contact is the sender	Clears out the content of all service request messages where the purged contact is the sender.
Clear SR Messages where the purged contact is a correspondent	Clears out the content of all service request messages where the purged contact is a correspondent.

1. Sign in as an administrator.
2. Navigate to the **Setup and Maintenance** work area.
3. Open the **Tasks** panel tab and click **Search**.
4. Search for and click the **Manage Administrator Profile Values** task.
5. On the Manage Administrator Profile Values page, navigate to the Search region.
6. In the Profile Option Code field, type `ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT` and click **Search**.
The `ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT` profile option is displayed in the Search Results: Profile Options region.
7. In the `ORA_SVC_SR_GDPR_OPTIONS_FOR_CONTACT`: Profile Values region, navigate to the **Profile Value** column for the Site **Profile Level**.
8. From the list of displayed values, select the value that you want to set for the profile option.

Audit Trail for Service Requests

How do I enable users to create an audit trail?

You can create an audit trail of changes made to service requests.

By default, only certain fields of the Service Request object are audited. If you want more fields to be audited, you must add them. You can also do an audit for the child objects by doing some configuration. Similarly, you must enable the audit for any administrator-defined child objects that you create from Application Composer.

To enable the audit for more fields or the child objects of the Service Request object:

1. Sign in to the application as a setup user or administrator.
2. Navigate to the Setup and Maintenance work area.
3. Open the Tasks panel tab and click **Search**.
4. In the **Search** window, search for and click the **Manage Audit Policies** task.
5. On the Manage Audit Policies page, click **Configure Business Object Attributes**.
6. On the Configure Business Object Attributes page, select **Service** from the **Product** drop-down list.

All the objects that can be audited are displayed. This includes the administrator-defined objects or administrator-defined child objects for predefined entities created using Application Composer.

For the Service Request object, the **Audit** check box is selected by default. The subset of attributes that can be audited are also displayed.

7. Click **Add**.
8. In the **Select and Add Audit Attributes** dialog box, select the check box next to the attribute that you want to add, and click **OK**.
9. To add attributes to audit for each predefined or administrator-defined child object of the Service Request object, select the check box next to the object.

Note: The predefined child objects and administrator-defined child objects of the Service Request object are displayed. For each of these child objects, one or more attributes are audited by default.

10. Repeat steps 7 and 8 to add more attributes.
11. Click **Actions > Synchronize**.
12. Click **Save and Close**.

Note: The audit for the Category object is enabled by default. But you can add more attributes by following the same steps as above.

FAQs About SR Management

Frequently Asked Questions (FAQs) for SR Management

This section contains some frequently asked questions (FAQs) when setting up and maintaining SR Management. Scroll or search to find answers to common questions.

What happens if two contacts are merged, and how does it impact service requests?

If two contacts are merged, you may see a change in the primary contact of a service request (SR).

This change occurs because the contact remaining after the merge is now associated with the SR, as it replaces the merged contact. However, no data is lost since the records are merged.

When two contacts are merged:

- The SRs associated with the merged contact are now associated with the remaining contact.
- If two contacts associated with the same SR are merged, you can see only the remaining contact in the SR.
- All the SR messages point to the remaining contact after merging.
- The SR shows all the interactions with both the merged contact and the remaining contact.
- Interactions for the remaining contact also show the interactions for the merged contact.
- Chat transcripts associated with both the merged and remaining contacts are associated with the remaining contact.

Note:

- You must select **All functional areas** as the merge scope when setting up Customer Hub using the Manage Customer Data Management Options task. It's necessary to select this merge scope for the service objects to be included in the scope.
- If you have defined an Application Composer rule that doesn't permit the primary contact on an SR to be updated, then the contacts don't merge.

Related Topics

- [Survivorship Rules](#)
- [How do I merge duplicate records?](#)
- [Automerger](#)

What happens if two accounts are merged, and how does it impact service requests?

If two accounts are merged, you may see a change in the account displayed in the **Account** field of the Summary subtab in a service request (SR). That's because the account remaining after the merge is now associated with the SR, as it replaces the merged account. However, no data is lost, as the records are merged.

When two accounts are merged:

- The SR objects associated with the merged account are associated with the remaining account.
- The merged account is no longer associated with the SR. Instead, the remaining account is associated with the SR.
- All the contacts associated with the merged account are now associated with the remaining account. So if the SR had a contact from the merged account, that contact is still associated with the SR.
- The SR shows all the interactions with both the merged account and the remaining account.
- Interactions for the remaining account also show the interactions for the merged account.
- Chat transcripts associated with both the merged and remaining accounts are associated with the remaining account.

Note:

- You must select **All functional areas** as the merge scope when setting up Customer Hub using the Manage Customer Data Management Options task. It's necessary to select this merge scope for the service objects to be included in the scope.
- If you have defined an Application Composer rule that doesn't permit the update of the account on the SR, then the accounts don't merge.

Related Topics

- [Survivorship Rules](#)
- [How do I merge duplicate records?](#)
- [Automerge](#)

Why are some service request reference numbers missing from the sequence?

Service request (SR) reference numbers aren't guaranteed to be in sequence. That's because they're generated as soon as the user clicks the Create Service Request button on the Service Requests page.

If the user decides to cancel the SR, then that SR reference number is lost because the same number isn't used again by the application.

SR reference numbers could be missing because of the following reasons:

- The SR reference numbers are based on a document sequence which has a cache size of 1000. So whenever a database server is restarted, it pulls the next 1000 numbers, even if the last set of 1000 numbers weren't all used up.
- If you have a couple of missing SR reference numbers within the set of 1000 numbers, the reason could be any of the following:
 - The SR was deleted.
 - An error occurred when the SR was being created.
 - The SR was canceled midway while creating it.

Why can't I set some service request fields by using groovy scripts or object workflow?

Sometimes, you may be unable to set a service request field by using groovy scripts or object workflow. If this happens, you must verify in the Application Composer whether it's a read-only field. You can't set a read-only field on a service request by using groovy scripts or object workflow.

4 Users and Security

Overview of Defining Setup Users

One of your first tasks when setting up the application is the creation of users who can perform setup tasks.

Oracle creates an initial user for you when your environment is provisioned. This initial user is configured to perform security tasks, such as creating other users and granting additional privileges. As an initial user you can create users, known as setup users, to help with application setup. The setup user performs the tasks in 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. You can access this task in the Setup and Maintenance work area by selecting these options:

- Offering: Customer Data Management
- Functional Area: Users and Security
- Task: Manage Users

Overview of Setting Up Users and Security

Since you followed the *Implementing Sales* guide steps to set up your initial set of users, then you already know that Oracle applications use a role-based access control security model to secure access to functionality and data.

In a role-based access control security model, users are assigned roles, and roles are assigned access privileges to protected system resources.

Sales and Service users who access the transactional UI, for example sales representatives or service representatives working in leads or service requests, are created as resources and are known as resources.

Default Preferences

To set up default preferences for users and roles, 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. See the *Securing Fusion Sales and Service* guide for more information.

User Identity Store

The Lightweight Directory Access Protocol (LDAP) identity store is a repository of user identity data. Your LDAP directory stores definitions of LDAP user accounts. In general, changes you make to user accounts are automatically synchronized between the sales application and your LDAP directory server. However, you must also run processes on a daily basis to manage the information exchange between your application and the LDAP directory server. For information, see the chapter about setting up application security in the *Securing Fusion Sales and Service* guide.

Setup Tasks in the UI and Other Setup Options

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

Setup Task or Option and Navigation	Description
Manage Job Roles Task Navigation: Setup and Maintenance > Sales Offering > Users and Security functional area	Oracle provides many predefined job roles. The relevant sales roles are listed in the <i>Implementing Sales</i> guide. You perform the Manage Job Roles task to: - Review the role hierarchy of a job or abstract role. - Create custom job and abstract roles. - View the roles assigned to a user and list the users who have a specific role. This task opens the Roles tab of the Security Console.
Manage Duties Task Manage Sales and Service Access Management Task Navigation: Setup and Maintenance > Sales Offering > Users and Security functional area	You perform the Manage Duties task to: - Review the duties of a job or abstract role. - Manage the duties of a custom job or abstract role. - Create custom duty roles. This task opens the Roles tab of the Security Console.
Manage Data Security Policies Task Manage Sales and Service Access Task Navigation: Setup and Maintenance > Sales Offering > Users and Security functional area	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. You can also use the Manage Sales and Service Access task to review and configure data security. This task opens the Sales and Service Access Management work area. For information, see the <i>Securing Fusion Sales and Service</i> guide.
Manage Users Task Navigation: Navigator > Users and Roles item or Setup and Maintenance > Sales Offering > Users and Security functional area	You create application users in the UI using the Manage Users task. A user with the IT Security Manager job role performs the Manage Users tasks. Note: You can also create sales users by importing users. For information on the user import options available, see the <i>Understanding Import and Export Management for Fusion Sales and Service</i> and <i>Implementing Sales</i> guides.
Manage HCM Role Provisioning Rules Task Navigation: Setup and Maintenance Sales Offering > Users and Security functional area	Oracle provides predefined role mapping rules for provisioning many of the standard job roles included with the application. However, using the Manage HCM Role Provisioning Rules task, you can create any additional role mappings you need to, to control the provisioning of roles to application users. For example, you can create a role mapping to provision the Channel Sales Manager role automatically to specific sales managers.
Import and Export Management	You can import users in bulk using data files. For information on the user import options available, see the <i>Understanding Import and Export Management for Fusion Sales and Service</i> and <i>Implementing Sales</i> guides.

Setup Task or Option and Navigation	Description
Import Partner Users Task	You can also import partner contact data using the Import Partner Users task. For more information, see the Getting Started with Your Partner Relationship Management Implementation guide.
Single Sign-On Authentication	Single sign-on authentication is optionally available for 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, and you 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 https://support.oracle.com .
Resetting User Passwords	Setup users 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 Implementing Sales guide for more information.
Updating Email Addresses	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're updating the email addresses of sales users, then you can also use the same import process you use to create them. See the Implementing Sales guide for more information.

Note: Other data security tasks listed in the Users and Security functional area task list don't apply to the sales applications. Follow the guidance in the [Implementing Sales](#) and [Securing Fusion Sales and Service](#) guides.

Related Topics

- [Set General Preferences for All Users](#)
- [Subject Areas for Adoption Reporting](#)

What are resources?

Here's an overview of resources and the different ways that 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. Use these predefined, out of the box, resource roles, or create your own resource roles.

Resource Directory

The Resource Directory gives you detailed information about all the resources within a deploying organization. You can also use the Resource Directory to find and communicate with other resources, and to network and collaborate with them. 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 you want to use them. For instance, you can assign resource organizations engaged in customer service activities to the Customer Service Organization usage. You can then 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 be made up of 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.

Setting Up Resource Teams

How do I manage resource teams?

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 can't be hierarchically structured and isn't intended to implement an organization.

Create Resource Teams

To create resource teams:

1. Navigate to the Manage Resource Teams UI page as follows: **Navigator > Resource Directory > Tasks > Manage 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**.

Edit Resource Teams

To edit resources teams:

1. Navigate to the Manage Resource Teams UI page as follows: **Navigators > Resource Directory > Tasks > Manage 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**.

How Resources 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.

How Resource Team Membership and Role Assignment Components 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 don't 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 **Navigators > 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 doesn't 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

Frequently Asked Questions (FAQs) for Resource Team

This section contains some frequently asked questions (FAQs) when setting up and maintaining a Resource Team. Scroll or search to find answers to common questions.

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 can't be hierarchically structured and isn't 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

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.

For a more complete list of job roles, see the Security Reference for Fusion Sales and Service guides on Help Center.

Job Role	Description
Customer Service Manager	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.
Customer Service Representative	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.
Knowledge Analyst	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.
Knowledge Manager	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.

In addition to the job roles, the following table lists the duty roles specific to the Service offering.

Duty Role	Description
Service Request Troubleshooter	Manages all aspects of the Service Request object. A customer service representative is typically given this duty role
Service Request Contributor	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.

Duty Role	Description
	This role is typically given to non-service users, for example, sales representatives, product managers and engineers.
Service Request Power User	Do all the tasks that a troubleshooter can, with the added privileges to manage queues. Customer service managers are typically given this duty role.
Service Request Administrator	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.
Service Request Channel User	Manages all aspects of a service request that is associated with a partner account. This duty is similar to the service request troubleshooter.
Service Request Partner User	Submits service requests to get assistance with partner sales or support issues. This duty is granted to a contact from a partner account.

Resource Roles and Provisioning Rules for Service

Oracle provides resource roles for the Service offering which are used to provision the standard service job roles. Oracle also provides the role provisioning rules for these resource roles so that service users are automatically assigned the job and abstract roles they need.

These are the service role provisioning rules provided by Oracle, the condition that triggers the provisioning, and the job and abstract roles each rule provisions.

Provisioning Rule Name	Condition	Job or Abstract Roles Provisioned
Service Vice President	HR Assignment is Active Resource Role is Service Vice President	Customer Service Manager Resource
Service Administrator	HR Assignment is Active Resource Role is Service Administrator	Customer Relationship Management Application Administrator Resource
Service Manager	HR Assignment is Active Resource Role is Service Manager	Customer Service Manager Resource
Service Representative	HR Assignment is Active Resource Role is Service Representative	Customer Service Representative Resource

Note: If you didn't use the Create Company Information quick setup task to set up your company information, then the predefined role provisioning rules aren't created; you must create the provisioning rules yourself. For information about creating provisioning rules, see the topic Create Rules to Automatically Provision Job Roles to Sales Users.

Related Topics

- [Overview of Role-Based Access Control \(RBAC\)](#)
- [How can I create rules to automatically provision job roles to sales users?](#)

Set Up Service Request Visibility Based on Queue

How do I set up request visibility based on queue?

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

Restricting access ensures users see only the SRs that are in their queue by completing the following processes in the Security Console:

- "Remove Data Security Policies from Users": If the users have been assigned other data security policies that grant them access to a larger set of SRs, then you must remove such data security policies from the users.
- "Assign Data Security Policies Based on Queue": Grant queue-based visibility to SRs for specific roles. This ensures that users with these specific roles can see only the SRs assigned to the queues where they're a resource member.

Remove Data Security Policies from Users

If your users have existing service duty roles and you don't want them to view all the service requests that they can currently view, 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 Fusion Securing Sales and Service* guide.

2. Remove the data security policies that you don't want these users to have.

For more information about removing or creating data security policies, see "Managing Data Security Policies" in the *Oracle Fusion Securing Sales and Service* guide.

Related Topics

- [Edit Data Security Policies on the Security Console](#)
- [Create Job and Abstract Roles](#)
- [Copy and Edit Duty Roles](#)
- [Copy Job or Abstract Roles](#)

Assign Data Security Policies Based on Queue

The predefined roles don't have queue-based service request (SR) visibility. Let's say you assign the queue-based data security policies to your users. Then, users with these specific roles can see only the SRs assigned to the queues where they're a resource member.

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 Fusion Securing Sales and Service* guide.

2. For the new job role created in the previous step, create a new data security policy by specifying the following attributes:
 - Give an appropriate name to the policy.
 - Select the following Data Resource: **Service Request Header**.
 - Select the following Data Set: **Select by instance set**.
 - Select the following 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**.
 - 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

- [Edit Data Security Policies on the Security Console](#)
- [Create Job and Abstract Roles](#)
- [Copy and Edit Duty Roles](#)
- [Copy Job or Abstract Roles](#)

Set Up Service Request Visibility Based on BU

How You Set Up Visibility Based on BU

When users view lists of service requests (SRs) 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.

You can restrict users' access based on their Business Unit (BU) membership, ensuring that the users see only the SRs that are assigned to the BUs where they're a resource member. To do so, you must complete the following processes in the Security Console:

- **"Remove Data Security Policies from Users"**: 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.
- **"Assign Data Security Policies Based on BU"**: Grant BU-based visibility to service requests for specific roles. This ensures that users with these specific roles can see only the SRs assigned to the BUs where they're a resource member.

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

Related Topics

- [Edit Data Security Policies on the Security Console](#)
- [Create Job and Abstract Roles](#)
- [Copy and Edit Duty Roles](#)
- [Copy Job or Abstract Roles](#)

Remove Data Security Policies from Users

If your users have existing service duty roles and you don't want them to view all the service requests that they can currently view, 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 Fusion Securing Sales and Service* guide.

2. Remove the data security policies that you don't want these users to have.

For more information about removing or creating data security policies, see "Managing Data Security Policies" in the *Oracle Fusion Securing Sales and Service* guide.

Related Topics

- [Edit Data Security Policies on the Security Console](#)
- [Create Job and Abstract Roles](#)
- [Copy and Edit Duty Roles](#)
- [Copy Job or Abstract Roles](#)

Assign Data Security Policies Based on BU

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

To assign the data security policies based on BU 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 Fusion Securing Sales and Service* guide.

2. For the new job role created in the previous step, create a new data security policy by specifying the following attributes:
 - Give an appropriate name to the policy.
 - Select the following Data Resource: .
 - Select the following Data Set: **Select by instance set**.
 - 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

- [Edit Data Security Policies on the Security Console](#)
- [Create Job and Abstract Roles](#)
- [Copy and Edit Duty Roles](#)
- [Copy Job or Abstract Roles](#)

Data Security Policies on SRs

How do I define access group-based data security policies?

As a Customer Relationship Management Application Administrator or Sales Administrator, you can define access group-based data security policies on service requests . Though predefined data security policies are available, you've the option to define more data security policies based on access groups.

As compared to SQL-based data security policies, access groups are another way of granting data permissions to users. Instead of creating security policies using SQL, you can use object-sharing rules in the sales and service access management UI to define your visibility criteria.

For more information about creating data security policies, see the Access Groups chapter in the Oracle Fusion: Securing Sales and Service guide.

If you're replacing existing SQL-based data security policies with policies based on access groups, then remember to set an end date for those policies.

How do I define system-access-group-based data security policies?

Let's say you're a Customer Relationship Management Application Administrator or IT Security Manager. Then you can use system access groups and predefined rules to support service request data security for your users.

Every standard job role provided by Oracle has a corresponding system access group. The predefined object-sharing rules assigned to each system access group provide the same access to service request data as provided by the standard job roles.

Note: System access groups provide you an alternative way to manage a user's access to service request data. You can also create access extension rules to extend the access provided by the predefined rules to related objects.

System access groups are active by default. But the predefined object-sharing rules associated with each system access group are inactive by default. So you must activate these rules before you can use them. The association between system groups and predefined rules is also disabled by default. So for each system group, you must also enable the predefined rules you want to apply to the group.

The following table lists the predefined object-sharing rules for system access groups along with their descriptions.

S. no.	Predefined rule name	This rule provides access to:
1	All Service Requests	All service requests.
2	All CRM Service Requests	Access to all CRM service requests.
3	CRM Business Unit Service Requests	All CRM service requests associated with the user's business units.
4	CRM Partner Service Requests	All CRM partner service requests.
5	CRM Business Unit Partner Service Requests	CRM partner service requests associated with the user's business units.
6	CRM Service Request Queue Member	All CRM service requests assigned to queues that the user is a member.
7	CRM Service Request Queue Member Hierarchy	All CRM service requests assigned to queues that the user's subordinates are a member.
8	CRM Service Request Team	All CRM service requests where the user is on the SR team.
9	CRM Service Request Team Hierarchy	All CRM service requests where the user's subordinates are on the SR team.

S. no.	Predefined rule name	This rule provides access to:
10	CRM Service Request Assignee	All CRM service requests where the user is the assignee.
11	CRM Service Request Assignee Hierarchy	All CRM service requests where the user's subordinates are the assignee.
12	CRM Service Request Creator	All CRM service requests where the user created the SRs.
13	CRM Service Request Creator Hierarchy	All CRM service requests where the user's subordinates created the SRs.
14	CRM Service Request Partner Account	All service requests where the SR is associated with the user's partner account.
15	CRM Service Request Contact	All CRM service requests where the user is an SR contact.
16	All HR Service Requests	All HR service requests.
17	HR Business Unit Service Requests	All HR service requests associated with the user's business units.
18	HR Service Request Queue Member	All HR service requests assigned to queues that the user is a member of.
19	HR Service Request Queue Member Hierarchy	All HR service requests assigned to queues that the user's subordinates are a member of.
20	HR Service Request Team	All HR service requests where the user is on the SR team.
21	HR Service Request Team Hierarchy	All HR service requests where the user's subordinates are on the SR team.
22	HR Service Request Creator	All HR service requests where the user created the SRs.
23	HR Service Request Primary Contact	All HR service requests where the user is the SR primary contact.
24	HR Service Request Assignee	All HR service requests where the user is the assignee.
25	HR Service Request Assignee Hierarchy	All HR service requests where the user's subordinates are the assignee.

End Date Predefined SQL-Based Data Security Policies

Every predefined rule was created to match an existing predefined SQL-based data security policy. So if you're enabling predefined rules, you must end date the matching predefined SQL-based data security policy.

Here's what you must do:

1. Navigate to Security Console.
2. Edit the role for which you're enabling predefined rules.
The service request data security policies are given to duty roles such as Service Request Troubleshooter, Service Request Power User, and Service Request Contributor. So, you would edit the roles that you copied from these predefined roles.
3. Go to Step 3, Data Security Policies, in the wizard where you're editing the role.
4. Find the matching SQL-based data security policy for your predefined rule.
5. Edit the data security policy and set a past date as the end date for the data security policy.

Note: Let's say you're using the predefined service duty roles such as Service Request Troubleshooter. Then you must first copy those roles because the predefined roles can't be edited.

Define Access Group-Based Data Security Policies for Messages

As a Sales Administrator or Customer Relationship Management Application Administrator, you can define access group-based data security policies on SR messages.

Access groups are an alternative way of granting data permissions to users, compared to SQL-based data security policies. You can use object-sharing rules in the Sales and Service Access Management UI to define your visibility criteria.

To define access-group-based data security policies for SR messages:

If you don't want to use an existing access group, complete the following tasks:

1. Create a new access group.
2. Add members to the access group.
3. Create an object-sharing rule to grant access to SR messages for all the access group members.

Note: For more detailed information about the above steps, see the Oracle Fusion Securing Sales and Service guide.

Related Topics

- [Create an Access Group](#)
- [Add Members to Custom Access Groups Using the UI](#)
- [Create Object Sharing Rules](#)

Define Access Group-Based Data Security Policies for Categories

can define access group-based data security policies on service categories.

Access groups are an alternative way of granting data permissions to users as compared to the SQL-based data security policies. You can use object-sharing rules in the Sales and Service Access Management UI to define your visibility criteria.

To define access-group-based data security policies for service categories:

1. Create a new access group if you don't want to use an existing access group.
2. Add members to the access group.
3. Create an object-sharing rule to grant the access group members access to service categories.

Note: For more detailed information about the above steps, see the Oracle Fusion Cloud Securing Sales and Fusion Service guide.

Related Topics

- [Create an Access Group](#)
- [Add Members to Custom Access Groups Using the UI](#)
- [Create Object Sharing Rules](#)

Enable Viewing of Updates to SRs Secured Using Access Groups in Near Real Time

You can access service requests (SRs) that are secured using access groups in near real time. When an SR is created or updated, users can access the new or updated SR immediately.

You don't have to run the Perform Object Sharing Rule Assignment Processing scheduled process for users to access the SRs.

To implement near-real-time processing for service requests, you must enable the following profile options:

- Near Real-Time Transaction Tracking for Access Groups Enabled (ORA_ZCA_ACCESS_GROUPS_NEAR_REAL-TIME_TRACKING_ENABLED)

This option is enabled at the site level by default.

- Common CRM Signals Active (ORA_ZCA_ENABLE_SIGNALS)

This option is disabled by default.

For more information, see *Configure Real-Time and Near Real-Time Access for Access Group Object Records* in the Oracle Fusion Securing Sales and Service guide.

Related Topics

- [How do I configure real-time and near real-time access for access group object records?](#)

Set Up Read-Only Access to the Service Request Header

How You Set Up a Role With Read-Only Access to a Service Request

Here's how you set up a role that has read-only access to a service request (SR), but still be able to add SR messages such as Customer Entry, Response, Forward to the SR.

To make the SR read-only for a user, remove the Edit Service Request privilege from the user. After you remove the Edit Service Request privilege from a user's role:

- The user won't be able to edit these elements of the SR:
 - SR contacts
 - SR team members
 - Add or delete attachments
- The user won't be able to take the following actions on the SR List pages:
- The following actions will be available through the Action Bar:
 - Copy
 - Internal Note
 - Response

Note:

- A user won't be able to search for Knowledge articles or link articles to the SR. But if the user has the privilege to compose messages on the SR, the user would be able to insert existing linked articles to an SR message.

Related Topics

- [Create Job and Abstract Roles](#)
- [Copy and Edit Duty Roles](#)
- [Copy Job or Abstract Roles](#)

How do I set up a role with read-only access?

Here's how you set up a role with read-only access to the service request (SR) header alone, or both the SR header and SR messages:

1. Sign into the application as an administrator.
2. Navigate to **Security Console**.

3. Copy a service duty role, such as **Service Request Troubleshooter** .
When copying the role, select the **Copy Top Role** option.
4. Remove the privileges that you don't want to assign, from the copied role.
5. Copy a service job role, such as **Customer Service Representative**
When copying the role, select the **Copy Top Role** option.
6. Delete the **Service Request Troubleshooter** child role from the copied job role.
7. Associate the copied version of the **Service Request Troubleshooter** duty role to the copied job role.
8. Associate the copied job role with the users who need read-only access to the SR header alone, or both the SR header and SR messages.

Set Up Read-Only Access to the Service Request Header and Service Request Messages

How You Set Up a Role With Read-Only Access to Service Request and Service Request Messages

Here's how you set up a role that has read-only access to the service request (SR) header and SR messages.

To make the SR header and SR messages read-only for a user, remove the following privileges from the user:

- Edit Service Request
- Compose Customer Message
- Compose Internal Note
- Compose Response

After you remove these privileges from a user's role, the following actions won't be available on the SR (in addition to the SR header read-only restrictions):

- Internal Note
- Response

Related Topics

- [Create Job and Abstract Roles](#)
- [Copy and Edit Duty Roles](#)
- [Copy Job or Abstract Roles](#)

How do I set up a role with read-only access?

Here's how you set up a role with read-only access to the service request (SR) header alone, or both the SR header and SR messages:

1. Sign into the application as an administrator.

2. Navigate to **Security Console**.
3. Copy a service duty role, such as **Service Request Troubleshooter** .

When copying the role, select the **Copy Top Role** option.

4. Remove the privileges that you don't want to assign, from the copied role.
5. Copy a service job role, such as **Customer Service Representative**

When copying the role, select the **Copy Top Role** option.

6. Delete the **Service Request Troubleshooter** child role from the copied job role.
7. Associate the copied version of the **Service Request Troubleshooter** duty role to the copied job role.
8. Associate the copied job role with the users who need read-only access to the SR header alone, or both the SR header and SR messages.

5 Set Up Activities Configurations

What Service Request Activities Track and Display

Activities track and display actions performed on individual service request records. The activities can include email exchanges, call logs, past and future tasks and appointments, as well as recent updates to the record itself.

Activities Configurations enable you to show only those objects, events and attributes that you want to display in the Activity Stream of your Service Center application.

Activities are designed to track important updates, but they don't function as an audit trail. For example, the list of activities for a service request may show that the SR was reassigned to a different agent. But the activities list may not show that someone uploaded an attachment or changed the value of an unimportant field. Administrators determine what information is displayed.

You can specify which attributes display in the Activities Stream, such as when certain values are updated, by creating an activities configuration and making it active. You create an activities configuration by duplicating an existing one.

Sprinkler system delivery: Activities

[Go to overview page](#)

Monday 7/11/22

-  **Victor Frances (CSS Automation Test8 US))** 12:58 AM
Service Request Contact added by Matt Hooper
-  **Matt Hooper updated Service Request** 12:58 AM
Primary Contact is updated to Victor Frances
-  **Matt Hooper updated Service Request** 12:02 AM
Title is updated to Sprinkler system delivery

7/7/22

-  **Matt Hooper updated Service Request** 11:19 AM
Account is updated to Pinnacle Technologies (SEATTLE, US)

7/6/22

-  **Comment** 7/6/22 7:03 PM 

Matt Hooper
Created 7/6/22 7:02 PM
Comment

← Reply

-  **Matt Hooper updated Service Request** 8:11 AM
Title is updated to xvdww

The screenshot shows activities and recent updates for a service request. The items shown are all the important changes Matt Hooper made to the SR. Each time you save, you create another entry with a time stamp.

What are the Activities setup tasks?

Here's a summary of the steps to set up activities in your environment.

Activities Setup Tasks

Step	Description	Navigation and other details
1	Make sure the profile option Set the Common CRM Signals Active (ORA_ZCA_ENABLE_SIGNALS) is set to No. This profile is set to the value you want by default, so you might be able to skip this step.	From Setup and Maintenance , choose Service . Then click the Tasks icon and select the Search link. Search for Manage Administrator Profile Values .
2	Run the Publish Feed Elements process to create the index required by Adaptive Search.	Navigator > Tools > Scheduled Processes <i>How do I run the Publish Feed Elements job to index activities?</i>
3	After the Publish Feed Elements process completes, set the Common CRM Signals Active (ORA_ZCA_ENABLE_SIGNALS) profile option to Yes.	From Setup and Maintenance , choose Service . Then click the Tasks icon and select the Search link. Search for Manage Administrator Profile Values .

Required Job Roles and Privileges for Creating Activities Configurations

To create activities configurations you need to have the following job roles and privileges.

Here are the required job roles:

- Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB)
- Application Implementation Consultant
- Customer Service Representative (ORA_SVC_CUSTOMER_SERVICE_REPRESENTATIVE_JOB)

Here are the required privileges:

Required Privileges for Creating Activities Configurations

Privilege Title	Privilege Code	Description
Manage Feed Element Generations	ORA_ZCA_GENERATE_FEED_ELEMENTS	Allows generation and publishing of feed elements.

Privilege Title	Privilege Code	Description
Access Feed View	ORA_ZCA_ACCESS_VIEW_FEEDS	Allows access to view feeds.
Access View Feed Configurations	ORA_ZCA_ACCESS_VIEW_FEED_CONFIGURATIONS	Allows access to view feed configurations.
Manage Feed Configurations	ORA_ZCA_MANAGE_FEED_CONFIGURATIONS	Allows create, delete and update of feed configurations.
Manage Signal Definitions	ORA_ZCA_MANAGE_SIGNAL_DEFINITIONS	Allows creating and updating signal definitions

How You Enable and Display Activities

You enable activities on the Configure Activities Display page.

Configurations generate and display content that you've chosen in the Activity Stream. By default, the provided configuration is activated. Create your configuration by duplicating the provided configuration and using it as a basis for your own activities configurations.

Let's look at the following screenshot to see the different parts of the Configure Activities Display page.

Configure What Displays in the List of Activities

What object do you want to configure? Object Service Request

Activities configurations (3/10)

Configuration Name	Active	Creation Date	Created By	Last Updated By	Last Updated Date	Action
Service Request Seed Config Clone	✓	2022-07-18	MHoope	MHoope	2022-07-18	...
SR Feed Config Cloned Copy		2022-06-06	SALES_ADMIN	SALES_ADMIN	2022-06-06	...
SR Feed Config Cloned Copy2		2022-06-07	SALES_ADMIN	MHoope	2022-07-18	...
Standard Service Request Activities Configuration		2021-03-11	SEED_DATA_FROM_APPLICATION	SEED_DATA_FROM_APPLICATION	2022-03-22	...

Configure : Service Request Seed Config Clone

Objects you can configure for display
Select an object to configure the display attributes.

- Service Request
- Action Plan
- Activity/Task
- Activity/Appointment
- Attachments
- Conversation Message

Object attributes available for display
ServiceRequest : Select attributes for display

Search for attributes by display name or unique name.

Selected Attributes (19/50)

Display Name	Unique Name	Make Available for Display	Monitor for Changes
Account ID	AccountPartyId	☐	☒
Account	AccountPartyUniqueName	☒	☐

Here's the different part is the Configure Activities Display page:

- **Callout 1:** Here's where you click the drop-down list and select the object you want to configure.
- **Callout 2:** Here are the list of available configurations. You can created 10 configurations per object. SEED_DATA_FROM_APPLICATION is the configuration that's provided by Oracle. You use this configuration as the basis for creating new configurations. You can't delete it, but you can duplicate and edit it.
- **Callout 3:** Click the **Action** field to display the various actions you can perform on the configuration, such as edit, activate delete, and duplicate.
- **Callout 4:** Here's where you can view an objects attributes.
- **Callout 5:** Here's where the selected and the available attributes for an object are displayed.

How do I create and activate a service request activities configuration?

You can create and activate an activities configuration for each object you use in Service Center. You can create an activities configuration by duplicating an existing one.

Oracle provides a predefined configuration for each object. You can use the predefined configuration or if you want to change what activities get displayed for an object, you can edit your copy before activating it.

You must have the Manage Feed Configurations duty role to can duplicate, edit, delete, and activate a configuration. Users with the Access View Feed Configurations duty role can only view the configurations.

Note:

- You can't edit or delete configurations provided by Oracle.
 - You can have only one active configuration for each object.
 - You can't delete an active configuration.
 - You can create a maximum of 10 custom configurations for each object.
 - You can enable Monitor for Changes for a maximum of 50 attributes for each object in a configuration.
-
1. Click **Navigator > Configuration > Configure Activities Display**.
 2. Select the **Service Request** object.
 3. In the Activities configurations section, select an existing configuration. Oracle provides one standard configuration for each object. Look for SEED_DATA_FROM_APPLICATION in the Created By column.
 4. In the Action column (the last column on the right), click **Actions** (three dots) and select **Duplicate**.
 5. In the Duplicate dialog, enter a name for your configuration and an optional description.
 6. Click **Duplicate**.
 7. Click on the new configuration name to display its attribute setup.
 8. If you want to use the configuration provided by Oracle, in other words, if you don't want to change any about the provided configuration you just copied, then skip steps 9 through 11 and activate the configuration as described in step 12.
 9. If you want to change the way attributes display first, then click **Actions** and select **Edit**.
 10. Configure the attributes:
 - a. Select one of the objects you want to configure for display on the left pane.
 - b. In the right pane, locate the attribute you want to update and set the two parameters:
 - **Make Available for Display**: makes a field available for display on the UI.
 - **Monitor for Changes**: monitors the field for updates.
 11. When you're done, click **Update**.
 12. To make your configuration active, click Actions (the three dots) and select **Activate**. You must then confirm that you want to activate.

Use **Actions** (the three dots) menu to activate, delete, duplicate, and update your configurations.

Before You Run Activities Initialization Jobs

Here are the jobs roles and privileges you must have to run this initialization job.

- Job roles with the Manage Feed Element Generations privilege can do the following:

Generate and publish activities which populate the activities stream data for users. The Sales Administrator, Service Administrators, and Application Implementation Consultant job roles have this privilege by default.

- User roles with the Access Feed View privilege can do the following:
Access the activities REST services to view the activities of the object. This Sales Administrator, Service Administrators, and Application Implementation Consultant job roles have this privilege by default..

You also want to make sure users meet the following prerequisites before invoking the activities initialization job:

- Make sure the job is available for the Sales Administrator and Application Implementation Consultant job roles.
- Once the job is run, refer to *How do I create and activate a service request activities configuration?*.
- Make sure the ORA_ZCA_ENABLE_SIGNALS profile option is disabled, as shown in *What are the Activities setup tasks?*.
- Generate import and export artifacts for the Activities-enabled objects then makes sure the generation of the artifacts is complete.

Specify What Displays in Activities

You can specify what information displays in Activities for each record by creating an activities configuration in the Configure Activities Display page and activating it. You must create and activate a separate activities configuration for each object.

In the configuration, you can make information visible by checking two check boxes:

- Make Available for Display: Displays the attribute in activities.
- Monitor for Changes: Tracks changes in the attribute.

You must check both check boxes for the information to be visible. For many attributes, you check both check boxes for the same attribute. For example, note text is stored in the NoteTxt attribute. So, to display notes salespeople create any subsequent updates to the note text, you select both check boxes for the NoteTxt attribute:

Attribute	Unique Name	Make Available for Display	Monitor for Changes
Note	NoteTxt	Yes	Yes

To display information from certain fields, including dynamic choice list fields and fixed choice list fields, you must check the check boxes for different attributes. That's because the information that displays in the UI comes from the name column while the tracking of changes comes from the ID column. For example, if you're displaying the owner of a task, a dynamic-choice list field, there are two attributes at play:

- Owner
- Owner Party ID

The Owner field stores the name of the person identified by the Owner Party ID. You want to display the owner name in the UI, but you have to monitor the Owner Party ID to see if a new owner has been assigned. Here are the settings you make for displaying the owner:

Attribute	Unique name	Make Available for Display	Monitor for Changes
Owner	OwnerName	Yes	No
Owner Party ID	OwnerPartyID	No	Yes

Some information displayed in this example, such as the postal address, come from aggregating multiple fields. The attribute FormattedAddress displays the address in the UI. FormattedAddress is composed of many elements including country, state, city, province, street address, and so on. There are different fields involved in different formats for each country, which you can review using the setup task Manage Formatted Addresses. Presumably, you want to display the changed address if any of address elements are updated. For that to happen, you must check the Monitor for Changes check box for all the address elements you're using and check the Make Available for Display check box for the FormattedAddress attribute. Here are the settings for just some of the available address fields:

Attribute	Unique name	Make Available for Display	Monitor for Changes
Address Line 1	AddressLine1	No	Yes
Address Line 2	AddressLine2	No	Yes
Building	Building	No	Yes
City	City	No	Yes
Country	Country	No	Yes
Postal Code	County	No	Yes
State	State	No	Yes
Address	FormattedAddress	Yes	No

If you check **Make Available for Display** for any of the address components, then you'd also see an entry for that component as well. Suppose, for example, you check **Make Available for Display** for City:

Attribute	Unique Name	Make Available for Display	Monitor for Changes
City	City	Yes	Yes
Address	FormattedAddress	Yes	No

The Activities would display two lines when the city is updated. For example:

- Address changed to 100 Main Street, New York, NY 10021
- City changed to New York

How do I run the Publish Feed Elements job to index activities?

During initial Service setup, the activities for all your existing records are blank. You must run the Publish Feed Elements job to populate activities for each record with the date and time of the record was created. The process also publishes the indexes required for the Adaptive Search to track activities for all the objects that have activities enabled. You can't run the process separately for individual objects.

Note: You run this process only once during initial Service Center setup.

1. Click **Navigator > Tools > Scheduled Processes**.
2. On the Scheduled Processes Overview page, click **Schedule New Process**.
3. In the Schedule New Process dialog box, leave the type as **Job**.
4. In the **Name** field, type **Publish Feed Elements**, and press **Enter**.
5. Make sure that the **Production to Test Enabled** checkbox is unselected. You use this option only to migrate the activity indexes from production to test as part of your data migration.
6. Continue to the next steps. Or, just skip to the steps for finishing the submission, to run the process once as soon as possible with the default output.

Which records get populated with activities?

The Publish Feed Elements process populates activities in records that meet the following criteria:

Object	Criteria
Account	All accounts of type Customer and Prospect with any tasks and appointments added in the last 300 days.
Contact	All contacts that were updated in the last 300 days.

The process also populates activities for child objects of the objects that meet the preceding criteria.

Account

Child Object	Criteria
Activity Type = Appointment	All activities of type Appointment that haven't passed the due date. All activities of type Appointment that have passed a due date 180 days from the current date.
Activity Type = Task	All activities of type Task that have the following status: Not Started, In Progress, and On Hold. All tasks in Completed status that has passed the due date 180 days from current date

Child Object	Criteria
Notes	All available notes that were created or updated in the last 180 days.
Audit	Audit log that shows the Created By and Updated By information for the Contact record.

The process also populates activities for child objects of the objects that meet the preceding criteria.

Contact

Child Object	Criteria
Activity Type = Appointment	All activities of type Appointment that haven't passed the due date. All activities of type Appointment that have passed a due date 180 days from the current date.
Activity Type = Task	All activities of type Task that have the following status: Not Started, In Progress, and On Hold. All tasks in Completed status that has passed the due date 180 days from current date
Notes	All available notes that were created or updated in the last 180 days.
Audit	Audit log that shows the Created By and Updated By information for the Contact record.

The process also populates activities for child objects of the objects that meet the preceding criteria.

Service Request

Child Object	Criteria
Audit	Audit log that shows the Created By and Updated By information for the service request record.
Messages	All service request messages of the following type: - Customer Entry - Response - Internal Note - Forwards - System Note - System Response
Interactions	Interactions for all SRs where StatusCd = "ORA_SVC_CLOSED" AND ChannelTypeCd IN ["ORA_SVC_CHAT","ORA_SVC_PHONE"]

Note: After this process has completed, you must set the ORA_ZCA_ENABLE_SIGNALS profile option to **Yes**.

Enable viewing of Activities records created through import management

Activity records created during the import management process aren't visible in the Activity panel. To enable these records to be viewed, do the following:

1. Add a custom field to the Activity object, such as a checkbox field called **UpdatedByImport**.
2. When importing, set a value of **Y** for this field.
3. Create a routine that runs on a schedule (such as hourly or daily based on the requirements)
 - a. The routine is based on the Activity object.
 - b. The action is field update: **Update the UpdatedByImport** to **N**.
 - c. Set the **UpdatedByImport** property of the rule set to **True**
4. Ensure the field UpdateByImport is configured to be available on the UI and then monitor the change in the Configure Activities display.

How do I migrate an index of activities from production to test?

If you want to migrate the index created for activities from your production environment to a test environment, you must run the Publish Feed Elements process in the production environment with the Production to Test Enabled check box selected.

Before you run the Publish Feed Elements process in the production environment, make sure you completed the following:

- You set up activities in the environment as outlined in the *What are the Activities setup tasks?*.
- You created an activities configuration in both production and test.
- You ran the Publish Feed Elements process in the test environment.

Note: If Data Masking is enabled during the Production to Test process, all your activities generated for the objects are deleted. You must run the Publish Feed Elements process again after the production to test process completes to populate the activity stream with the masked data.

Related Topics

- [How do I run the Publish Feed Elements job to index activities?](#)
- [Create and Activate an Activities Configuration](#)

6 Extend Service Center

How do I extend my Service Center application?

Your application is built using Oracle Visual Builder Studio and Oracle JET components. Visual Builder Studio is a single place where organizations can manage projects, environment and repositories; a Visual Builder Designer is where you can extend your application. From VB Studio, you can open many different designers.

Visual Builder Studio is the browser-based tool that your organization can use to make modifications and extensions to the Redwood user interface. Using the extension framework, you can make changes to the user interface such as displaying custom fields on forms, creating dynamic layouts for a form, or embedding custom content in a dynamic container in a page.

Your VB Studio instance is paired with your Test instance of Fusion Applications. This means that your design time experience will be based on data in the Test instance database and a sandbox in the Test instance (if you're working within a sandbox).

The changes you make using VB Studio are stored in an artifact called an application extension. An application extension can be something as simple as a field added to a cloned, custom, or new form, or a new container of forms. You can create extensions, then preview them to see how they look in your UI. To create a extension that impacts the data model, you must use a sandbox within Application Composer, then use the Edit Pages in Visual Builder option to expose your changes in the UI.

You navigate to the VB Studio Designer directly from within the Redwood application. VB Studio is a **design at runtime tool**, which allows you make UI changes and then immediately preview the results before publishing the changes to other users.

VB Studio is just one tool that administrators can use to make application changes. As previously mentioned, you extend the underlying object model used by the Redwood application with Application Composer. The model changes you make in Application Composer are reflected in the RESTful API for those objects, and thereby become available to the VB Studio Designer for use in UI extensions.

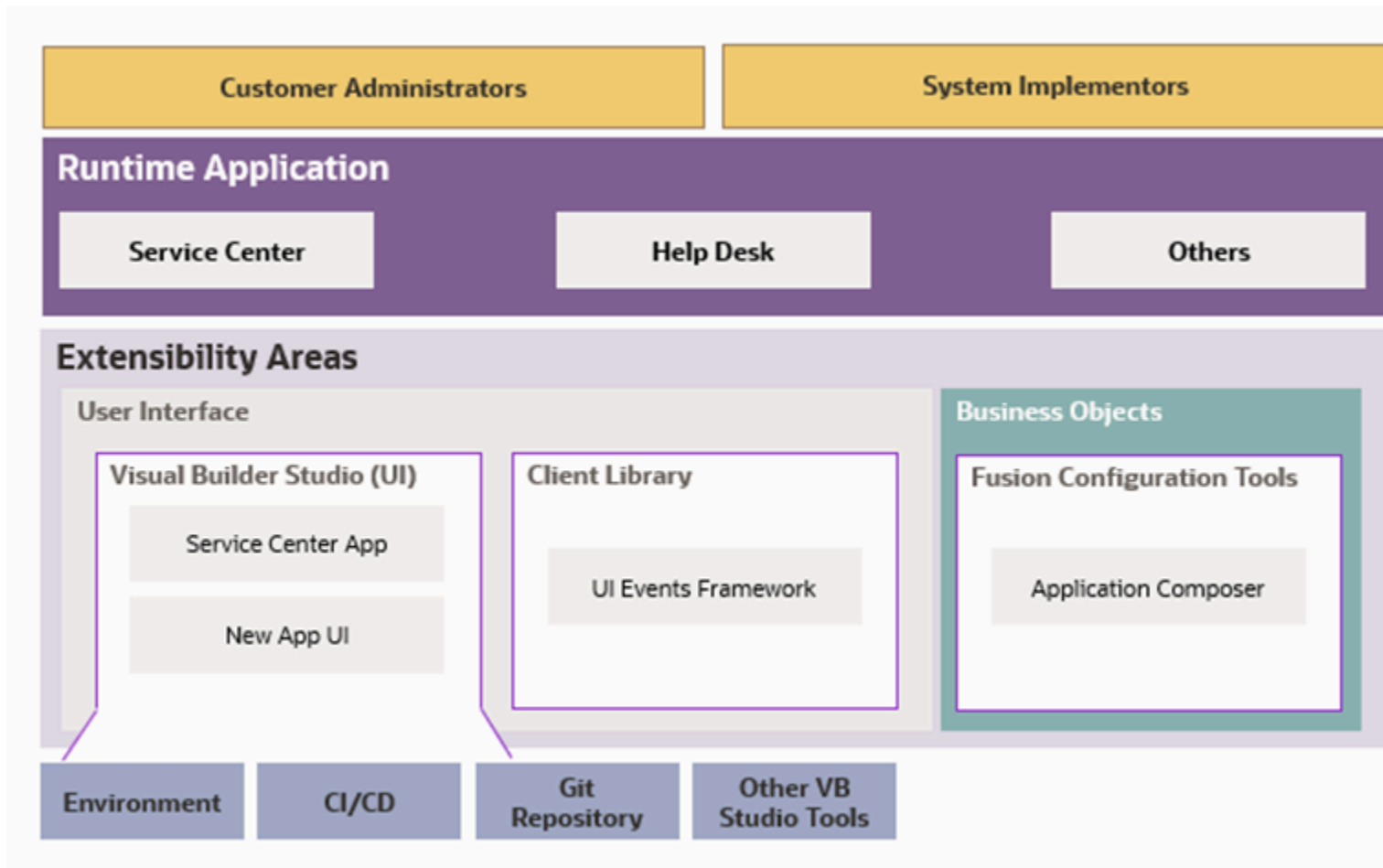
Related Topics

- [Configuring Applications Using Application Composer](#)
- [How Do I Use Visual Builder Studio to Extend Oracle Cloud Applications?](#)

Extensibility at a Glance

While Oracle Fusion Service Center applications provide robust ready-to-use functionality, you might want to extend certain parts of your application to meet your business needs.

Here's an overview which describes the various components to consider when extending your application.



Here's a brief explanation of the elements shown in the graphic:

- **Applications and Components:** These are the main components you'll work with when extending the application.
- **Visual Builder Studio:** Enables you to create your pages and flows through the App UI, or ready-to-use pages using Oracle CX Service UI Extension App.
 - Note:** The App UI is part of the Visual Builder Studio, and the core features are common among all Fusion Applications or nonFusion Applications.
- **UI Events Framework:** A client-side library that you can use in third party applications. Using this library, you can create bidirectional communication between Service Center and the application. For example, say you have an in-house order management system that needs to be embedded in Service Center and listen to events such as save or change field value, so the third party application can implement a logic upon Service Center events. Or, from the third party application, fields in the Service Center can be populated and reduce agents copy and paste activity while an embedded application.
- **Application Composer:** Use Application Composer to extend objects (add, remove or change fields) and create smart actions that initiate events from the server.

Set Up Visual Builder Studio

Before your team can start creating application extensions, you must first set up Visual Builder Studio and configure it.

Note: It's strongly recommended that you've only one deployed extension. Having multiple extensions can lead to changes being lost or not implemented. Having all changes in one extension streamlines future updates and enables you to have a complete picture of work that has been done.

Get Started

Start with *Essential Set-up Tasks* in the Administering Visual Builder Studio guide.

Set Up VB Studio

Next, read the following instructions in *Set Up VB Studio to Extend Oracle Cloud Applications* in the Administering Visual Builder Studio guide.

This chapter tells you how to set up VB Studio so your users can work with Oracle Cloud Applications in and extension.

How to Provision a VB Studio Project Using an Application Template

Before you extend your application, you must first provision a VB Studio project and use the Service Center Extensibility Template.

Here are the steps to provision a project and get Visual Builder ready:

1. Access Visual Builder for your instance.
2. Click **Create**.
3. In the **New Project** dialog box, enter the project's name, and then click the **Private** or the **Shared** button in the Security area depending on your needs.
4. Click **Next**.
5. Select **Application Extension** and click **Next**.
6. Click the drop-down list for **Oracle Cloud Applications Development Instance** and select your instance, then you base application and template.
For the Service Center, select **Oracle CX Service UI Extension**.
7. Click **Next**, then click **Finish**.

Service Request Standard Pages Extensibility Matrix

Service Center offers numerous extensible components in its out-of-the-box pages making it possible to adopt them and accelerate your project implementation. The following table shows a matrix of Service Center standard pages along with their extendable components along with possible usage of those components. Use this matrix to address your business needs using the out-of-the-box pages.

- Customize Dynamic Tables and Forms are forms and tables used in our standard pages that you can extend. For example, create and change a layout such as adding and removing fields.
- Custom Actions are containers in certain pages that enable you to add new buttons to perform validation and actions on top of the standard pages.
- Display Your Own Content refers to containers that enable you to create your own content, such as a new foldout panel.
- Configurable Components comprise variables which can be exposed in the standard pages to change an extension's behavior. For example, the back button link on the service request detail page can be changed so the extension can pass a new label name and URL that goes to a custom app instead of back to the Service Request list.
- The UI Events Framework is a client-side library that you can use to extend an external application to listen to Service Center events or initiate actions creating a seamless integration and a new level of extension.

Page Name	Customize Dynamic Tables and Forms	Custom Actions	Display Your Own Content	Configurable Components	UI Events Framework	Possible Usage
landing	No	Yes	Yes	Yes	No	• Create and change the layout (for example, add and remove fields including custom business object). • Create a field template (for example, input text make it as a hyperlink). • Add buttons to perform actions and validations. • Add and remove columns from Service Request list. • Add a row-level action to the Service Request list.
create	Yes	No	No	No	Yes	• Create and change the layout (for example, add and remove fields including custom business object). • Create a field template (for example, input text make it as a hyperlink).
edit	Yes	Yes	Yes	Yes	Yes	• Create and change the layout (for example, add and remove fields including custom business object). • Create a field template (for example, input text make it as a hyperlink).

Page Name	Customize Dynamic Tables and Forms	Custom Actions	Display Your Own Content	Configurable Components	UI Events Framework	Possible Usage
						- Add buttons to perform actions and validations. - Change the label and URL path of the link to return to the list page. - Modify variables using an extension.
copy	Yes	No	No	No	Yes	- Create and change the layout (for example, add and remove fields including custom business object). - Create a field template (for example, input text make it as a hyperlink).
svc-interaction	Yes	Yes	Yes	Yes	Yes	- Create and change the layout (for example, add and remove fields including custom business object). - Create a field template (for example, input text make it as a hyperlink). - Modify variables using an extension.
categories	Yes	No	No	No	No	- Add and remove columns from the category list. - Add and remove fields from the search form. - Create rule set for dynamic UI.
search-results	No	No	Yes	Yes	No	- Change the search container - Modify variables using an extension.
view-article	No	No	Yes	No	No	Modify variables using an extension.

Extend Service Center navigation

Use the Service Center Extensibility Framework to extend your Service Center user experience.

As an administrator, you can use the `cxSvcNavigation` event, to listen to navigation events across the Service Center application, and use actions to enable navigation to the Service Console tab.

Here are the actions:

- Open Service Console Tab and Close Service Console Tab actions are used for custom page navigation.
- Open Page Inline and Close Page Inline actions enable custom secondary tab navigation.
- Set Service Console Tab Properties action can be used to update tab labels, icons, and other attributes.

Here's an example of how you implement them:

1. In Visual Builder Studio, go to **Service** within the Customer Experience (CX) group.
2. Open the service application and click the **Event Listener** tab.
3. Create a new `cxSvcNavigation` listener and click the **Create Action** chain.
4. Add an `if` condition and test it using one of the following payload attributes:
 - `application`: string
 - `navigationType`: string
 - DRILLDOWN
 - SCREENPOP
 - `flow`: string
 - `page`: string
 - `pageParams`: object
 - `engagementData`: object
 - `extraAttributes`: object
 - `sourcePageInfo`: object
5. From the Action Palette, find "CX Service: Open Service Console Tab" and add it to the Action Chain.
6. Define the properties to navigate directly to the details page:
 - Application: service
 - Flow: sr
 - Page: edit
 - Page Input Parameter: { `srNumber: event.pageParams.srNumber, view: 'detail' }`}
 - Tab Label: { `{ event.pageParams.Title }` }
 - JET CSS Icon Class: `oj-ux-ico-accessor-more`

Note: Set the `stopPropagation` property to True when you want to prevent the standard behavior.

Make sure you have a targeted condition so the action will not fire in all pages, unless that's your intention. In the previous example, if you want the custom navigation to happen only on the Service Center home page, you'd create this statement:

```
if (event.sourcePageInfo.page === "service-home" && event.application === "service" && event.page === "edit")
```

Disable the Service Center dashboard button

The Service Center dashboard shortcut appears by default next to the Create Service Request button on the Service Request list page. Though this shortcut icon gives agents a quick way to access the Service Center dashboard, it might not fit with your implementation.

Here's how you can disable it:

1. From the Service Request list page, click the Settings and Actions menu, and select **Edit Page in Visual Builder Studio**.
2. Select a workspace, then Oracle CX Service UI Extension App.
3. Expand Service Center, then **service > ec > sr**.
4. Click the **Variables** tab.
5. In the Constants list select **showDashboardButton**.
6. In the Constant properties pane, click the **Default Value** drop-down list and select **false**.
7. Preview your changes, then publish.

Modify any field on posted service request messages in Redwood

You can allow users who've been assigned the correct privilege to edit all fields on any SR message of any type (except system generated messages), including the PII checkbox.

Posted SR messages often need to be edited to fix typos, errors or to tag them as containing sensitive PII information. To improve data quality administrators can grant users the ability to change posted SR messages. Here's how you do it.

- If you want a user to be able to modify the message body to remove PII data or fix errors in any posted SR message, grant the user the SVC_EDIT_SR_MESSAGES_PRIV privilege.
- If you want a user to be able to modify the SR message header, to deselect the PII checkbox, in any posted SR message, grant the user the SVC_EDIT_SR_MESSAGE_HEADERS_PRIV privilege.

Users with these privileges can now edit all SR messages no matter the message type (except system messages) or creator of the message.

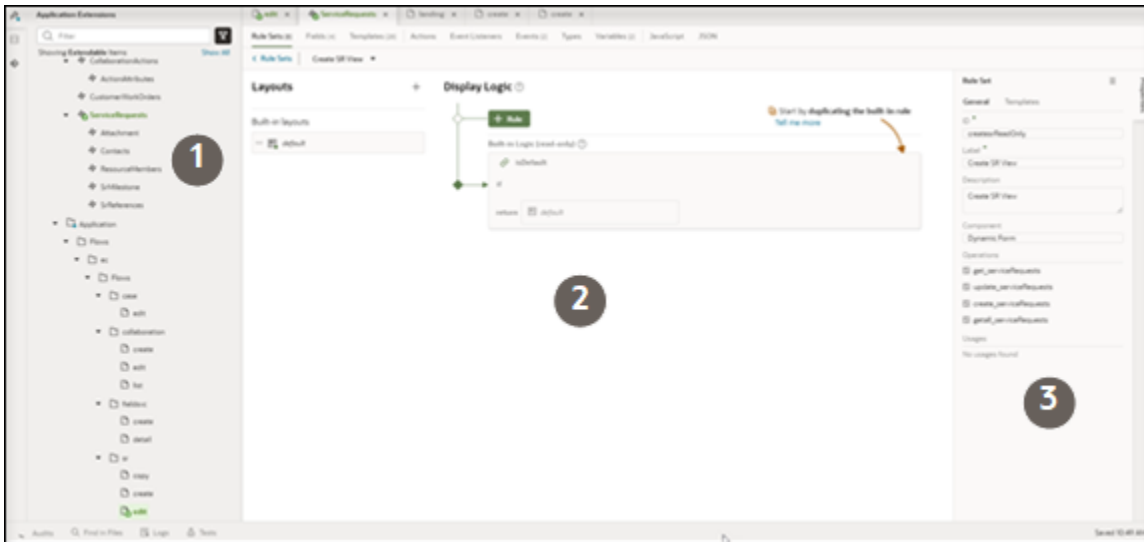
Work with Dynamic UI

Get to Know Layouts

After you launch VB Studio, VB Studio Designer displays the form you were examining in Service Center.

You can extend components within the Create Service Request page, the Edit Service Request page, and other pages within the application. For this example we'll work with the **Create Service Request** page.

The following image shows the various components of the VB Studio UI:



Here's the various elements of the UI:

- The navigator (callout 1): Use the explorer to view your project artifacts.
- The canvas (callout 2): Use the workspace to view your forms and containers.
- The Properties pane (callout 3): Use the Properties pane to view and change properties.

There are two ways to navigate to dynamic UI:

Method One: Directly Editing the Dynamic UI

Here's an overview of how you edit directly in the dynamic UI.

1. When you're on a Redwood page, such as Create Service Request, click the Settings and Actions menu and select **Edit Page in Visual Builder**.
2. In VB Studio, expand **App UIs > Oracle Fusion Service UI Extension App > Service > ec > sr**.
3. All the sub-components are displayed under the service requests component.

Note: This is where you'd go to extend the service request header dynamic forms. .

Method Two: Using the Page Designer

Here's an overview of how you navigate layouts using the Page Designer.

1. In Service Center, click the Settings and Actions menu and then select **Edit Page in Visual Builder**.
2. Choose your project and click **Select**.

The Page Designer displays.

3. In the application extension explorer, expand **Oracle Fusion Service UI Extension App > service > ec > sr > create**.

The Create Service Request form displays in the Page Designer with the corresponding artifacts are shown in the Components area of the Properties pane.

4. Click the **serviceRequests - Create Service Request** component.
5. Click the **Create Service Request** rule set link.
6. In the Properties pane, click the **Open Rule Set Editor** link.

The Rule Set is displayed. The Rule Set is where you define how the fields in a given component will appear at runtime.

7. Duplicate the rule set., then click the **Open** icon.

Extensible parts of the form are highlighted and you can view the corresponding component in the Properties pane.

Related Topics

- [What are Dynamic Components?](#)
- [Understand the Designer](#)

What dynamic forms can I extend?

The Service Center UI is highly extensible. This topic gives you an overview of what layouts in the Service Center Service Request UI then you can extend using Visual Builder Studio.

The following two tables list many of the dynamic forms that you can configure, and many of the dynamic tables, used for service requests, that you can extend.

Dynamic Form Name	Description
Create Service Request	Use this to extend the Create Service Request form.
Copy Service Request	Use this to extend the Copy Service Request form.
Service Request Spotlight	Use this to extend the SR Summary Region of the SR.
Service Request Contact Summary Layout	Use this to extend the SR contacts section in the SR Summary Region the SR.
View Service Request Contact Collapsed Layout	Use this to extend the Contacts Summary form in the related panel section of the SR.
View Service Request Contact Expanded Layout	Use this to extend the Contacts Details form in the related panel section of the SR.
Service Request Attachments Layout	Use this to extend the SR Attachments section in the SR Summary Region of the SR.

Dynamic Form Name	Description
Service Request Team Summary Layout	Use this to extend the Service Request Teams section in the SR Summary Region of the SR.
View Service Request Team Layout	Use this to extend the SR Team Members dynamic form in the related panel section of the SR.
Service Request Linked Articles Summary Layout	Use this to extend the SR linked articles section in the SR Summary Region of the SR Details form.
Service Request Details (View Mode)	Use this to extend the read-only SR Details form.
Service Request Summary (Edit Mode)	Use this to extend the editable SR Details form.
Service Request Source and History (Edit Mode)	Use this to extend the editable form of the Source and History section in the SR Details form.
Category Details	Use this to add custom fields the Categories List form.
Create Category	Use this to add custom fields the Create Category form.
Update Category	Use this to add custom fields the Update Category form.

Dynamic Table Name	Description
Edit Service Request Contact Layout	Use this table to extend the SR Contact list in the SR details page.
Team Section Table Layout	Use this table to extend the SR Teams list table in the SR details page.

Work with Dynamic Forms

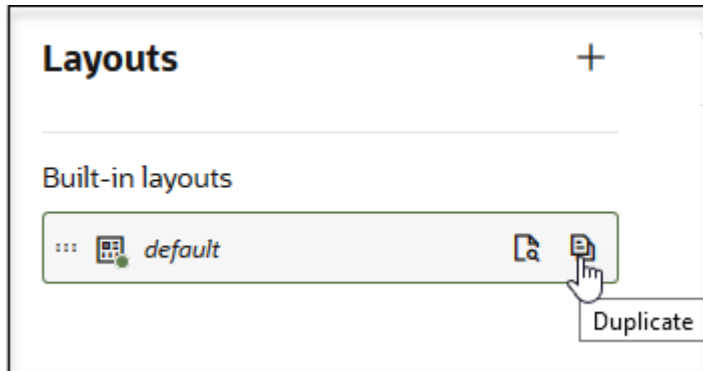
How do I duplicate a form in Visual Builder?

This example, using the Create Service Request form, will show you how to extend or change a form.

Note: It's highly recommended that you duplicate any form that you work with before you change or extend it.

1. Click the App UIs tab, then **Oracle Fusion Service UI Extension App > service > ec > sr > create**.
2. In the Properties pane, click the **Components** tab, and select **serviceRequests - Create Service Req**, and then click the **Open Rule Set Editor** link.
3. In the Designer, you'll see the available layouts, along with the display logic of that layout.

4. Select a layout.
5. In the Built-in layouts area, hover in the default layout, and select the **Duplicate** button.



6. In the **Duplicate Layout** dialog box, enter a name for the layout and click **Duplicate**.

View Form Properties

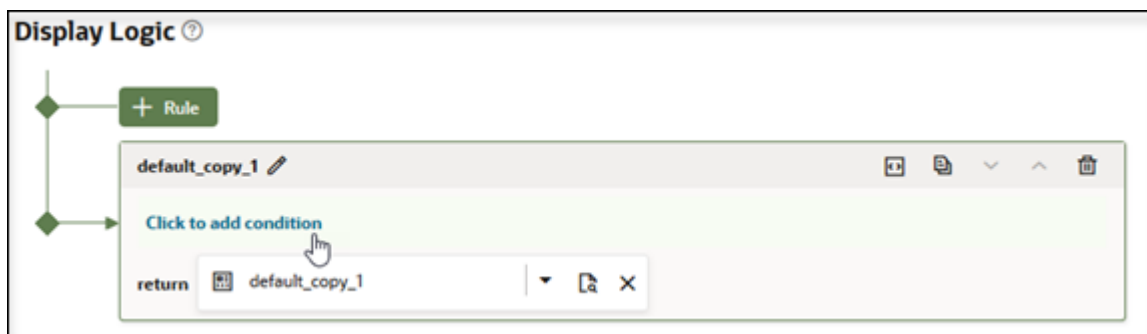
Now let's just look at some properties of the new form. The Rule Set properties list the form's properties along with a list of templates.

1. After you duplicate the form, select it in the Designer.
2. Click the **Fields** tab to view available fields for the form.

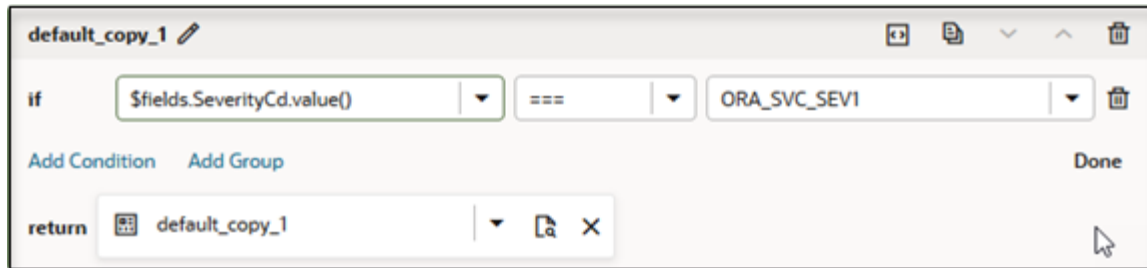
How do I add conditions to a form using VB Studio?

In this example, we'll add a condition that specifies that a form displays only if a certain SR severity level is reached. First duplicate the layout, or choose your layout, then do the following:

1. Click the **Click to add condition** link.



2. From the **Attribute** drop-down list, select the severity code value: `$fields.SeverityCd.value()`, then in the Value field, enter: ORA_SVC_SEV1 as shown in the following example:

The screenshot shows a rule set editor window titled 'default_copy_1'. It contains a conditional rule with the following structure: 'if' followed by a dropdown menu containing '\$fields.SeverityCd.value()', then an equals sign '===', and another dropdown menu containing 'ORA_SVC_SEV1'. Below this, there are buttons for 'Add Condition' and 'Add Group'. At the bottom, there is a 'return' section with a dropdown menu containing 'default_copy_1' and a 'Done' button.

Now the layout will only display when a service request has a severity code value of 1.

Example of Making a Layout Visible Based on Category Conditions

Here's an example of how to make your layout visible when using category conditions:

1. In the categoryId field add the value within single quotes as follows:
`$fields.CategoryId.Value() === '123'`
2. For the categoryName field to display, you must expose it in one of the create layouts.
3. Create a dummy layout and expose **categoryName** and use the category Name value in the conditional rendering.

For example:

```
$fields.CategoryName.Value() === CategoryName Document URL and section
```

How do I add, hide, or rearrange fields?

This is a worked example on how you work with form fields in VB Studio. You can add, hide or rearrange the order of fields. As with previous examples, we'll use Create Service Request.

1. In the Rule Set editor, from the Dynamic Form list, select the **Create Service Request** form.
2. Duplicate the layout, then select the duplicate record.
3. Click the **Open** icon.

The layout is displayed with its included fields, and in the explorer, all available fields.

4. To add a field, click the check box of any field shown in the field explorer.

The added fields, including custom fields added with Application Composer, will appear at the end of the layout.

5. To reorder fields, click the control beside the field, and drag the field to where you want it to go in the layout order.

The following screen capture shows the control you use to drag and drop fields in the layout.



6. To hide a field either deselect the field in the field list, or hover in the field in the layout and click the **X**.

How do I create a custom field?

Create a custom field in Application Composer if you want changes retained to the database. Custom fields created in VB Studio are read only.

This example shows you how to create a custom field using Application Composer.

We'll create a new field for the Service Request **Category** object. Say you want to extend this object with a custom field that's specific to your business needs. In addition to creating the custom field in Application Composer, you'll also need to add the custom field to the appropriate layout which is described in the following section. When both of these steps are performed, then the new field can be exposed in the Manage Service Request Categories List page.. You might also want to make the content entered into this custom field searchable. Here's how you do it:

Note: You'll need to create and activate a sandbox to perform this task. Use the same sandbox used for this task when you add the field in VB Studio, which is described in the next task.

1. Sign in to your application as an administrator.
2. Click **Navigator > Configuration > Application Composer**.
3. Expand the **Standard Objects** list, then expand the **Category** object node.
4. Click **Fields**.
5. On the Fields screen, click the **Custom** tab, then click the **Action** drop-down list and select **Create**.
6. Select **Text** because for this example, we'll just create a text field.
7. Click **OK**.
8. In the Create Text Field screen enter the field name into the **Display Label** field.

The Display Label field displays the field name in the UI. The Name field auto-fills after you complete the display label, and is only for internal use. It won't be displayed in your UI.

9. Choose the number of characters you want to allow in the new text field.

The default number of characters is 30, but you can set it for less or for more.

10. If you want the field to be searchable, select the **Searchable** check box in the Constraints area.
11. Click **Save and Close**.
12. To add the field in VB Studio, follow the steps in the next procedure.

How do I add a custom field created in Application Composer?

Use this topic to add a custom field you created in Application Composer. Fields that you create in Application Composer allow you to retain changes to the database, whereas custom fields created in VB Studio are read only.

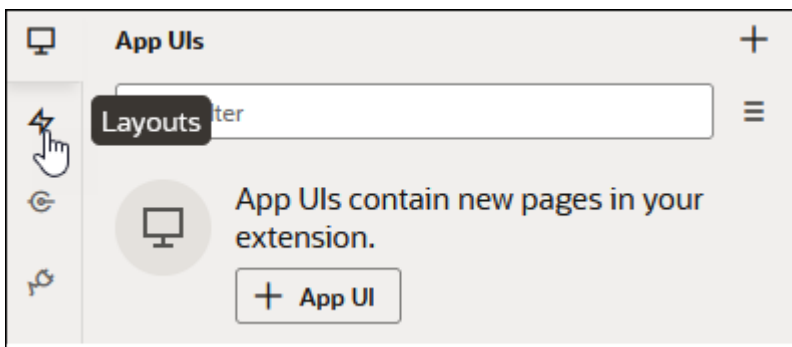
This topic shows you how to add a custom field that you created in Application Composer to a layout in VB Studio.

Before you start

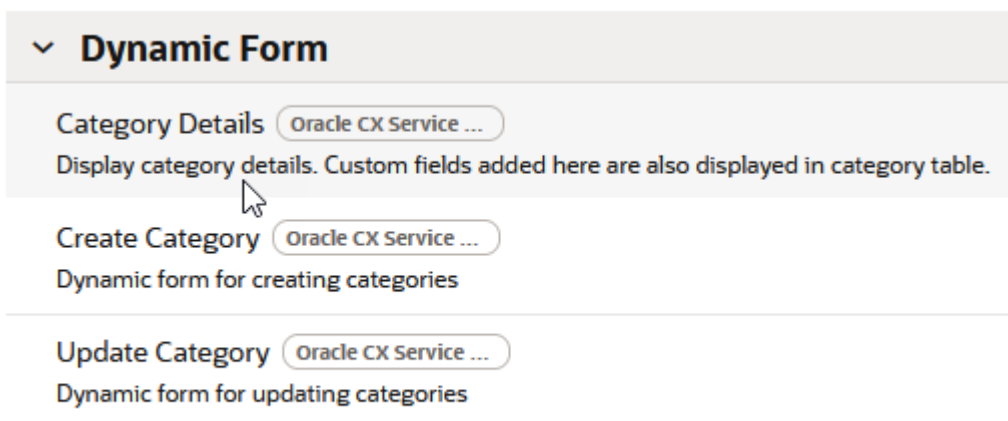
Make sure VB Studio is connected to the same sandbox that you used when creating the custom field.

Here's what to do

1. In VB Studio, click the **Layouts** icon in the explorer.



2. Find the object you want to add the field to.
3. In this example, we'll use the **Categories** object, so click that object.
4. Display the rule sets, and then choose the one you want to add the field to.
5. For this example, choose **Category Details**.



6. Duplicate the form using the following task: [How do I duplicate a form in Visual Builder?](#)
7. Open the duplicate you created.
8. The workspace displays the fields that are contained in the form, and those that can be added.

9. From the list of available fields, find your new field, using the filter field to search if necessary.
10. Drag the new field to the form, and position it in the order you want. You can use *How do I add, hide, or rearrange fields?* for reference.
11. The field is now added to the form. Click the **Preview** button to view your changes in the application UI.

How do I add a read only custom field using VB Studio?

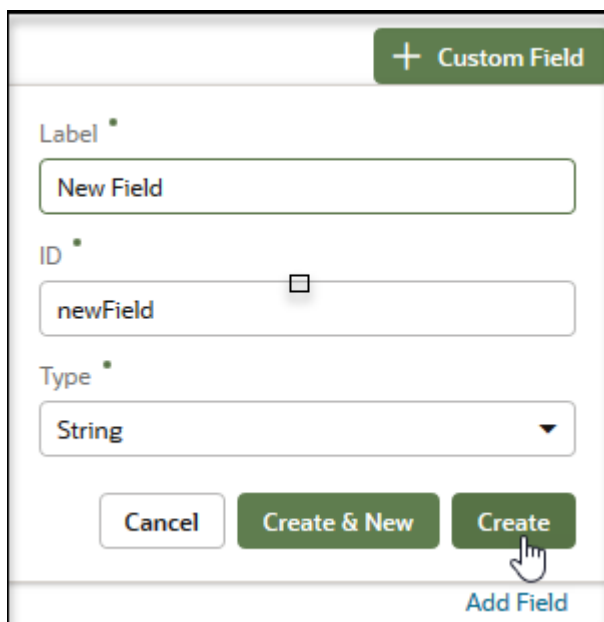
Now, we'll create and add a custom field in VB Studio then add it to a layout. For this example, we'll be creating a custom field based on the SR severity.

Note: Custom fields created in VB Studio are read only. To create a custom field that will retain changes to the database, you must use Application Composer. To create a custom field in Application Composer, see *How do I create a custom field?*

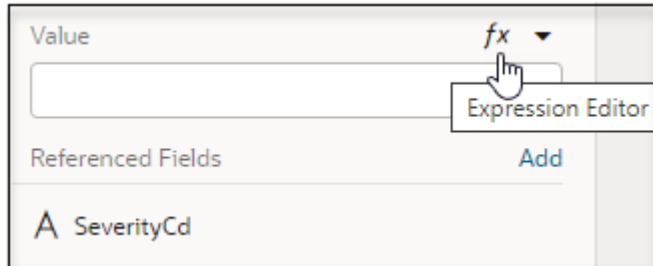
1. In the Applications Extensions explorer, navigate to **App UIs > Oracle Fusion Service UI Extension App > Service > ec > sr > edit**.

Remember, we're using the Service Request layout for this exercise.

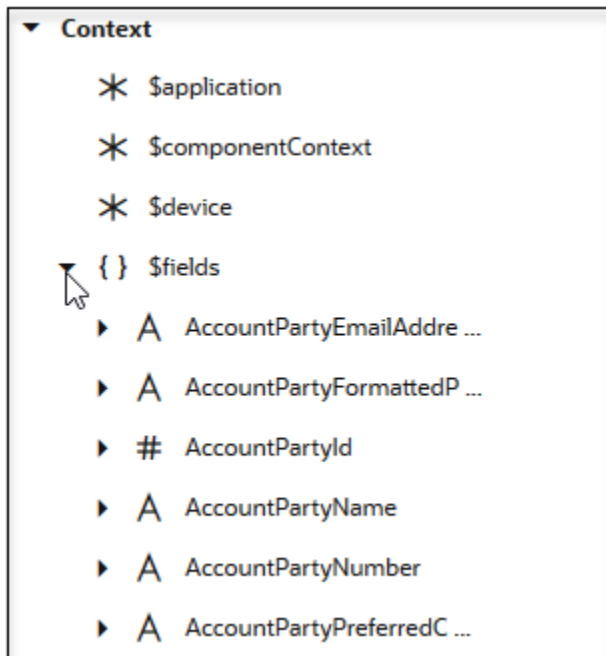
2. Click the **Fields** tab.
3. Click the **+ Custom Field** button.



4. In the **Custom Field** dialog box, enter the field name that you'd like reflected in the UI.
5. Click **Create**.
6. From the list of fields, select the newly created custom field.
7. In the Properties pane, add referenced fields by doing the following:
 - a. Click the **Add** link for the Referenced Fields property.
 - b. In the **Referenced Field** text field, start entering the word and then choose the **SeverityCd** field from the results that appear and then click **Add Field**.
8. Now, click the Expression editor:



9. Now in the Expression Editor, set the conditions that govern when the field is displayed by doing the following:
 - a. From the Context explorer, expand the fields list.



- b. Find the **SeverityCd** field in the list, expand it, and drag **value** into the editor.
 - c. Add the following to the expression: `=== 'ORA_SVC_SEV1'? 'High Severity SR': 'Not a high Severity SR'`
- d. Click **Save**.

```
1 $fields.SeverityCd.value()=== 'ORA_SVC_SEV1'? 'High Severity SR': 'Not a high Severity SR'
```

Now the custom field will only display if and value has a severity code of 1.

10. Click **Rule Sets** to return to the layout.
11. In the Fields explorer, enter the name for the custom field.
12. Click the checkbox to include the field in the layout.
13. Drag the field to where you want it to appear in the UI.
14. Use the Field properties window to choose one of the default templates from the **Template** drop-down list.
15. Click the **Create** link to create your own template.
16. Click the **Preview** button to view your changes.

Details on creating a template are in the [Create Your Own Field Template](#) topic in this chapter.

How do I make a field read only or required?

This task shows you how to set properties on a dynamic form to make its fields read only or conditional. For this example, we'll use the Create Service Request layout.

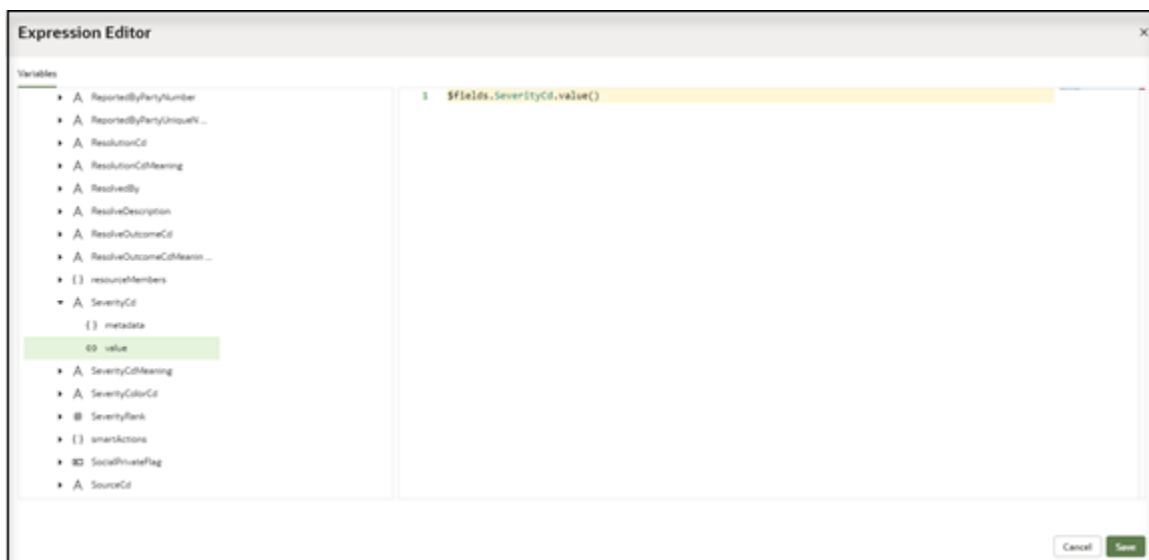
Note: Required and Read Only are only available if a field in a layout isn't associated with a template.

First duplicate the default form if necessary, or choose your own layout, and then do the following:

1. Select the layout.

The layout is displayed with the fields shown.

2. Note the **Read Only** and the **Required** check boxes in the Form properties area.
3. To make the field read only, simply click the **Read Only** checkbox.
4. To make the field required, just click the **Required** checkbox.
5. To make a field conditionally required, do the following:
 - a. Hover over, and then click the **Expression Editor**.
 - b. Select a variable from the drop-down list.
 - c. To edit the conditions of the variable, hover over then, select the **Expression Editor**.
 - d. In the Expression Editor, expand the **fields** variable list.
 - e. Expand the SeverityCd variable and drag **value** on to the editor workspace as shown in the following example:



- f. Add the following value to the expression: `=== 'ORA_SVC_SEV1' .`
- g. Click **Save**.

Now the field only displays when severity level 1 is reached

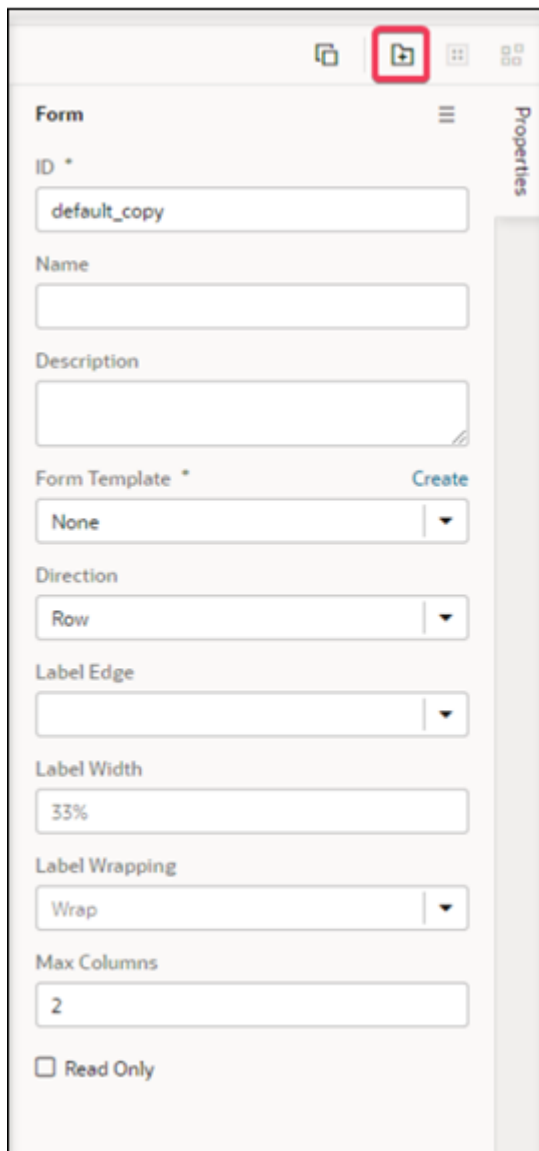
Field Groups

Field groups are a way for an application developer to group multiple fields. A field group can have properties such as column span, maximum columns, label, and so on. These properties apply to the whole group.

Field groups can also be conditionally hidden. These conditions apply to all the fields in the group.

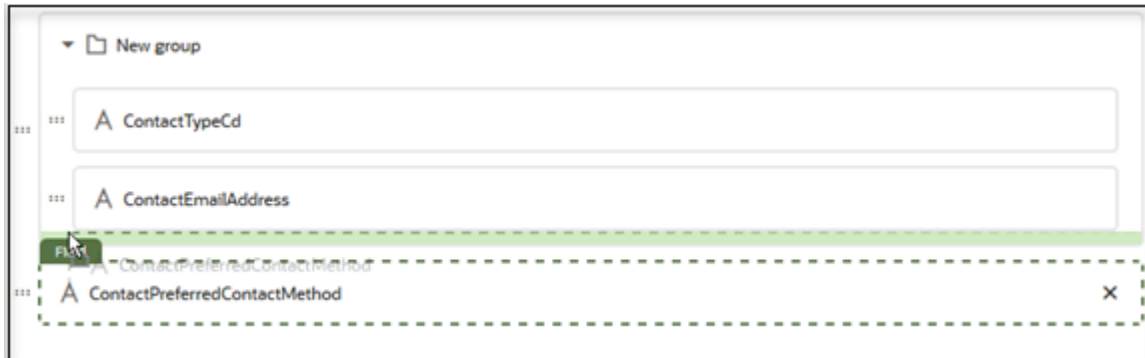
To create a field group, just do the following:

1. Select a **Dynamic form** layout.
2. Click the **Open** icon for the layout.
3. Click the **New Group** icon as shown in the following graphic.

The image shows a screenshot of the Oracle Fusion Service 'Form' editor. At the top, there are several icons: a copy icon, a 'New Group' icon (a square with a plus sign inside), a grid icon, and a list icon. The 'New Group' icon is highlighted with a red rectangular box. Below the icons, the 'Form' editor is visible, showing various fields and properties. The fields include 'ID *' (with a value of 'default_copy'), 'Name', 'Description', 'Form Template *' (with a value of 'None' and a 'Create' button), 'Direction' (with a value of 'Row'), 'Label Edge', 'Label Width' (with a value of '33%'), 'Label Wrapping' (with a value of 'Wrap'), and 'Max Columns' (with a value of '2'). There is also a 'Read Only' checkbox at the bottom. On the right side, there is a 'Properties' panel.

The field group is added to the layout.

4. Enter a name for the new field group, then click the check mark.
5. In the Properties window, enter values for the label of the group along with other properties such as maximum columns and `colspan` which determines the width of columns in your layout.
6. Add any fields you want into the group by dragging and dropping them from the explorer:



Create Your Own Field Template

You use field templates to configure how certain fields are rendered at runtime in a dynamic form or table.

You can use a combination of HTML and VB components to control how you want to display a field. A simple example for this would be: You want to display a red banner on the top of the SR if the SR is marked critical.

Here's how you do it:

1. Duplicate the **Service Request Summary (View Mode)** dynamic form (if you haven't done that already.)

Note: This form is the read only SR details form that you see when you click on the SR details icon on the SR summary region.

2. Since we want to display the banner at the top of the form, let's move the `criticalFlag` field to the top of the layout.
3. In the field properties pane, click the **Override** link.

This link lets you override the field template of the `criticalFlag` field. In other words, the override makes the template property editable. A Create link is also now visible.

4. Click the **Create** link to create the new template.
5. Enter a name for the template in the **Label** field. The **ID** field will auto-populate.

VBStudio designer opens. This is where you can design the template.

6. Click **Create**.

Note that there are tabs called Live, Design and Code. For this exercise we'll click on **Code** tab.

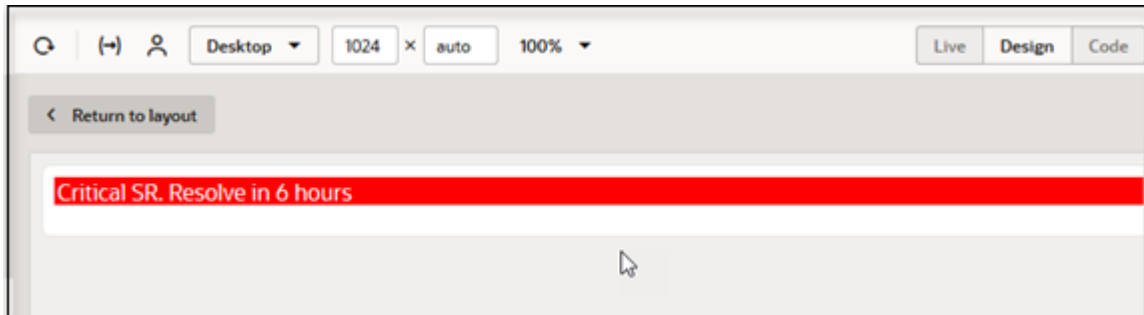
If you go to the Design tab, you can use the editor to drag and drop VB components to define your template.

7. Click **Code**, and note that VBStudio has created a default template for you. We are going to remove the code that's inside your template tag, and paste the following code inside the template tag:

```
<of-bind-if  
test="[[ $value]]">  
<oj-label-value colspan="6">  
<div slot="value" style="background:red">  
<span id="criticalBadge" style="color:white;padding:2px">
```

```
</of-bind-text value="Critical SR. Resolve in 6 hours">
</oj-bind-text>
</span>
</div>
</oj-label-value>
</oj-bind-if
```

8. Note that we've used the `<ob-bind-if>` VB element, `<oj-label-value>` VB element, the `<div>` HTML element, and the `<span>` HTML element.
9. Now, click the **Design** tab, and then click Return to layout.
10. Note the `CriticalFlag` is now associated with the newly created template.



Work with Dynamic Tables

Extend Dynamic Tables

You might want to extend a dynamic table, such as the SR Contacts table or the Team Members table to add or remove new fields or other elements.

Let's say that you want to remove the **Job Title** field and add the **DoNotContactFlag** field to the **SR Contacts** Dynamic table. Here's how you do it:

1. In the Application Extension explorer, click **Layouts**, then expand **Oracle Fusion Service UI Extension app > ServiceRequests > Contacts**.
2. Click the **Edit Service Request Contact Layout** dynamic table.

The layout rule set editor appears.

3. Duplicate the default layout if you haven't already done so.
4. Open the copied layout.

The layout editor appears displaying all the attributes currently displayed in the SR contacts table.

5. Delete the Job Title field by hovering over the field and clicking the **X** icon.
6. Use the explorer to locate the **DoNotContactFlag** field in the list of available fields.
7. Select the field by clicking the check box.

The field is added to the table. You can now move the field to where you want it to reside in the table.

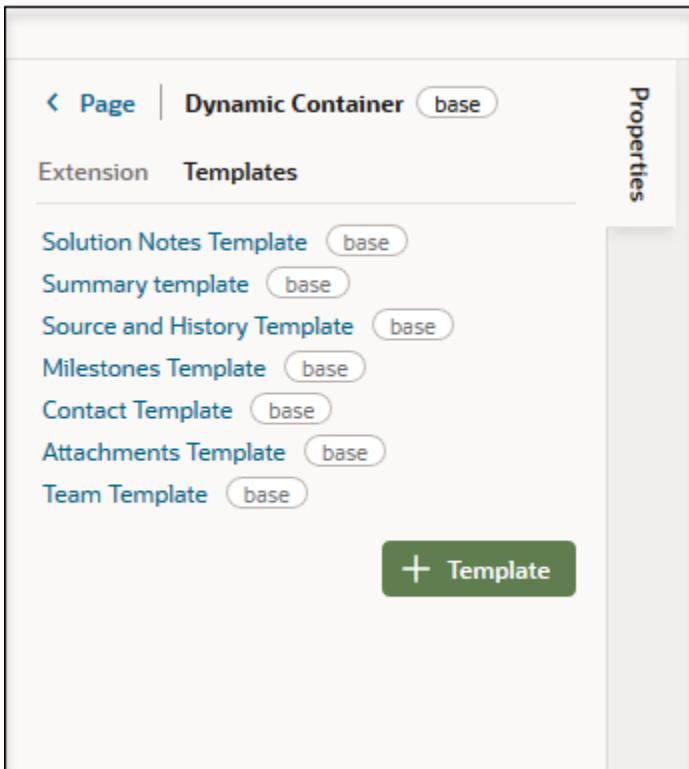
8. Test your changes by clicking the **Preview** button.

Work with Dynamic Containers

What's a Dynamic Container?

A dynamic container displays content in individual sections or logical regions of the page.

Each section maps to a template, which defines the content displayed within that section. Within Service Center, the **Edit SR Details** page has a dynamic container. The SR Edit page has multiple sections and templates in it:



The display logic for determining what's displayed in a dynamic container is defined using cases. A case is similar to the rule set layouts used in dynamic forms and tables, but instead of selecting which fields to display, you select which UI elements or components to display. When you define a case, you specify the conditions for the case, and the templates you want displayed in the container when that condition is met. The template defines the content you want to display. Each template in a case is rendered as a section in the container, so if a case defined three templates, you would see three sections in the container.

A dynamic container holds a collection of display templates. The display logic for determining what's displayed in a dynamic container is defined using cases. A case is similar to the rule sets used in dynamic forms and tables, but you select display templates containing UI elements instead of selecting fields. You configure what's displayed in a container by creating cases, and then specifying the conditions and templates for each case.

Related Topics

- [Configure a Dynamic Container](#)

Configure a Dynamic Container

The SR Edit Details page has seven sections. Let's say we have a requirement for a specific role. We don't want to show the Attachments section (Attachments template) to anyone with the Sales Representative role. Here's how you do it:

1. In the explorer, navigate to: **App UIs > Oracle Fusion Service Extensions App > Service > ec > sr > edit.**
2. In the properties pane, click the **Components** tab to view the dynamic container, form, and table used.
3. Click **Container rule for case details panel.**
4. Click the **Duplicate** icon.
5. Enter a name for the new rule.
6. Click the Expression editor icon and add the following condition:
`$application.user.roles.includes('ORA_ZBS_SALES_REPRESENTATIVE_JOB')`.
7. Delete the **documentsSectionTemplate. Attachments Template** from the case.
8. Test to make sure a sales representative won't see the attachments component in the **SR Edit Details** form

Note: The other details in VB will remain the same.

Work With Templates in a Container

Now we'll look at an overview of containers.

You can easily relocate or remove a template from a container. You can also add templates to the default set. You can also have more than one container for a given page layout and add conditions for the page layout by user role.

Move or Remove a Template in a Container

Let's first look at how you move or remove a template in a container.

For this example, we'll use the Service Request dynamic container.

1. In the VB Studio explorer, navigate to **App UIs > Oracle Fusion Service UI Extension App > Service > ec > sr > edit.**
The dynamic container is displayed.
2. In the Dynamic Container explorer, you can view the following:
 - Extension: View the different layouts contained in the container. Each case represents a layout.
 - Templates: View the different templates in the container view.
3. Choose a case from the list, and in the display templates area, hover over a template.
4. Move a template by clicking the up or down arrow.
5. Remove an item by clicking the **Remove Item** icon.
6. Preview the layout by clicking the **Layout Preview** icon:

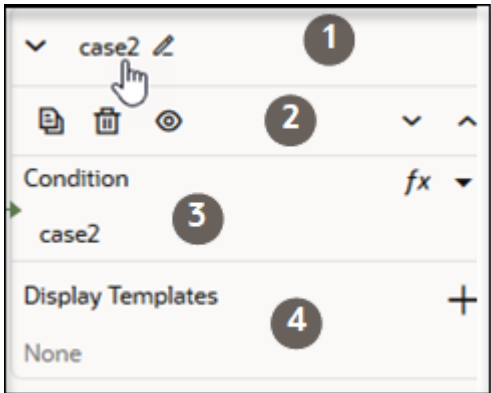


7. Add, relocate or remove templates as you see fit.

Add a Container

Now we'll add a new case, or container then add templates to it.

- 1. In the Dynamic Container explorer, click the **+ Case** button.
- 2. A new case appears in the properties page similar to the following:



The case properties are as follows:

Callout Number	What You Can Do
1	Click the Edit icon to create your own title.
2	Hover over each icon to view the task you can perform. ○ Duplicate the container. ○ Delete the container. ○ Preview the layout. You can also click the up or down arrows to move the position of the container in the layout.
3	Click the Expression Editor icon to write an expression for the container, or click the Select Variable drop down list and choose from a list of out of the box variables.
4	Click the Add Template icon (+) to select from a list of provided templates.

Service Request Categories

Manage Parent and Child Service Request Categories

Use the Service Request Categories setup page to move child categories between different parents, and promote child categories to a parent level. Also, you can use different attributes to search the categories.

Here's how you create service request categories:

1. Navigate to **Home > Service > Service Request Categories** (quick action).

The Categories page is displayed. This page is extensible, so you can also add custom fields.

2. Create a top-level category:

- a. Click **Create**.

The **Create Category** dialog box is displayed, with the **Create Top-Level Category** option selected.

- b. Enter a name in the **Category Name** field.
- c. Modify the **Category Code**, if needed.
- d. From the **Business Unit** drop-down list, select the business unit (BU) to which you want to associate the category.

The **Business Unit** field is displayed only if the HZ_ENABLE_MULTIPLE_BU_CRM profile option is enabled. By default, the BU that's set in the scope is displayed. However, you can change the BU.

- e. Specify if the category must be active, by using the **Active** switch.

The **Active** switch is enabled by default.

- f. Click **Create**.
- g. Create additional top-level categories, as needed.

3. Create child categories:

- a. From the categories list, select the top-level category for which you want to create child categories.
- b. Click **Create**.

The **Create Category** dialog box is displayed.

- c. Select the **Create Child Category** option.
- d. Enter a name in the **Category Name** field.
- e. Modify the **Category Short Name**, if needed.
- f. Specify if the category must be active, by using the **Active** switch.
- g. Click **Create**.
- h. Create additional child categories, as needed.

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

Note: You can set the BU only for the top-level category. The BU on the child categories is automatically set based on the BU of the root category.

Here's how you update existing service request categories:

1. From the categories list, select the child category you want to update, and then click **Update**. The **Update Category** dialog box is displayed.
 - To make a child category a parent category, use the **Promote to Top-Level Category** switch.
 - To move a child category to a different parent category, select a new category from the **Parent Category Name** drop-down list.
 - To modify the parent or child category name, update the name in the **Category Name** drop-down list.
 - To set the category as active or not, use the **Active** switch.
2. Click **Update**.

Here's some other tasks you can do:

- To see only the active categories on the page, use the **Show Active Only** switch.
- To search for a category, specify the search criteria under the **Search Categories** section, and then click **Search Categories**.
- To delete a category, select the category and click **Delete**.

UI Events Framework

Introduction to the UI Events Framework

The UI Events Framework is a JavaScript framework that you can use with a third-party application to interact with the Fusion application.

The framework has dedicated APIs to listen to events generated from the the Fusion application and perform operations inside the Fusion application. The UI Events Framework manages these events and operations. For example, an input Text Field in the Fusion application can be updated from a third-party application through a UI Events Framework operation named `setFieldValue`. You can also subscribe to events such as field-value changes, so that whenever the field's value change occurs in the Fusion application, the callback function registered from the third party application is executed with the new and the old values of the field. More examples of what UI Events Framework can do follow.

Architecture Details of the UI Events Framework

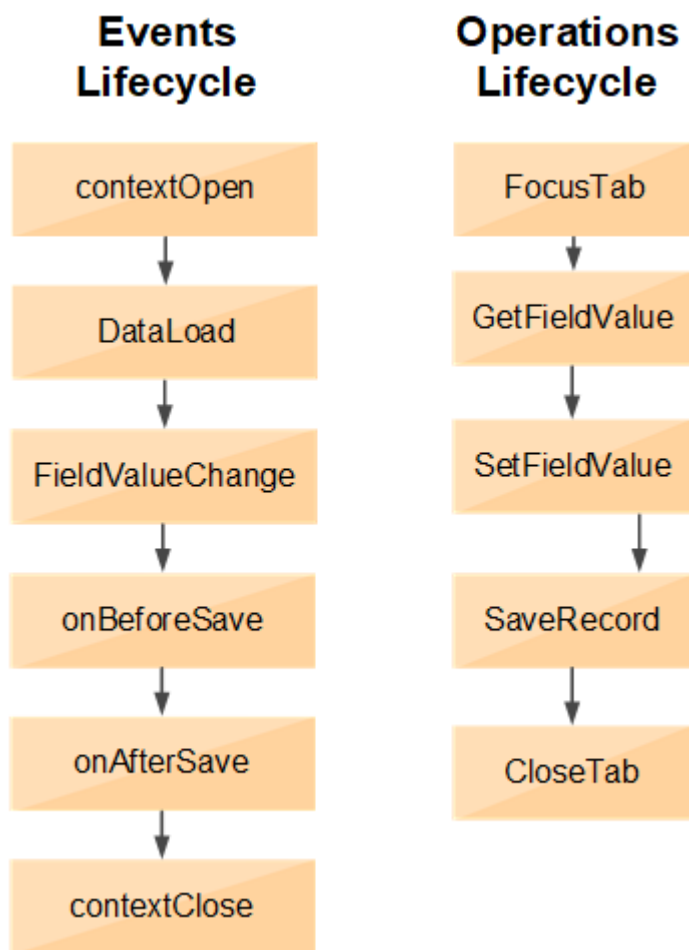
UI Events Framework can:

- Subscribe to events from Service Center.
Supported events include: `TabOpen`, `TabClose`, `ContextOpen`, `DataLoad`, `FieldValueChange`, `OnBeforeSave`, `OnAfterSave`, `ContextClose`, `TabChange`.
- Publish an operation to Service Center.
Supported operations include: `FocusTab`, `GetFieldValue`, `SetFieldValue`, `SaveRecord`, `CloseTab`.

Understanding the UI Events Framework life cycle

Here's an overview of the Events life cycle.

The following graphic presents a life cycle overview:



Set Up and Try UI Events Framework

In this section, you'll learn how to set up a new Customer App and integrate it into the Fusion application.

This section covers only a basic setup, along with an example of a basic publish operation and a subscribe to an event use case.

Here's what we'll look at:

1. Setting TypeScript in your third-party application.
2. Add UI Events Framework library to your third party application.
3. Load the your application in the Fusion application.

4. Subscribe to a Tab Change event.
5. Publish a Set Field Value operation.

Set Up Your Third-Party Application

This example uses TypeScript to extend the JavaScript.

Make sure the `Node.js` file is installed in your local development directory. The commands mentioned (in bold) can be executed in VS Code terminal.

1. Create a project folder and open it in any IDE - Recommended VS Code.
2. Initialize an npm project and install TypeScript and HTTP-server. `npm init -y`
3. Install TypeScript transpiler and HTTP-server globally. `npm install tsc http-server -g`. 'http-server' is used for local deployment.
4. Now run `Create tsconfig.json`:

```
{
  "compilerOptions": {
    "target": "es6",
    "module": "commonjs",
    "strict": true,
    "outDir": "out",
    "sourceMap": false,
    "lib": ["es2015.promise", "es6", "ES2016", "dom"]
  }
}
```

5. Now run `Create main.ts`.

```
function run(name: string) {
  console.log('Hi ', name)
}
```

6. Transpile TypeScript to JavaScript using `Command: tsc -w`. This creates a file `main.js` in the out file. (Notice that `typedefinition:string` in `main.ts` is removed in `main.js`).
7. Create a file `index.html`. Notice that in `index.html`, the `main.js` file is loaded from "out/" directory, which will be generated during transpile.

```
<html>
<head>
  <script src="./out/main.js"></script>
</head>
<body>
  <button onclick="run('uef') ">run</button>
</body>
</html>
```

The host `index.html` and `out/main.js` and go to the hosted URL in the browser.

Add the UEF Library to Your Application

The library name is `ui-events-framework-client.js`

A client application can be embedded in the Fusion application in 4 different ways:

- Load the client application in an iframe
- Custom `.js` in-app pages
- JavaScript in custom pages

- Custom Component CCA

Here's an example of how you load the application in an iframe.

You add the `ui-events-framework-client.js` file to the script section of `index.html` file.

```
<head>
<script src="https://static.oracle.com/cdn/ui-events-framework/libs/ui-events-framework-client.js"></script>
</head>
```

Load the Third-Party Application Inside the Fusion application

You use the `<oj-cx-svc-common-ui-events-container>` component to load external applications in the Fusion application. You install the component from Oracle Component Exchange and configure it in the Fusion application according to the instructions in the included Readme file. Its `url` attribute should point to the application that we've hosted locally. It can be hard coded or added through page-level variables.

Here's an example:

```
<oj-cx-svc-common-ui-events-container url="[[ $page.variables.url ]]"></oj-cx-svc-common-ui-events-
container>
```

Subscribe to the Tab Change Event

Update the `run()` function in `main.ts` file with code to handle the Tab Change event. Now the code will look like the example which follows. More details are available in description of the Subscribe API in *Call the APIs*.

Add `uiEventsFramework.d.ts` to `main.ts` as shown in the following example:

```
Update the d.ts file https://docs.oracle.com/en/cloud/saas/fusion-service/fuief/set-up-and-try-ui-events-framework.html#Subscribe-to-the-Tab-Change-Event . This link has mention to uiEventsFramework.d.ts. Link it with the following content
// uiEventsFramework.d.ts
declare namespace CX_SVC_UI_EVENTS_FRAMEWORK {
  enum EventName {
    'ContextOpen',
  }
  enum OperationName {
    GetFieldValue = 'GetFieldValue',
    SetFieldValue = 'SetFieldValue',
  }
}
let uiEventsFramework: IUiEventsFramework;

export interface IUiEventsFramework {
  initialize(applicationName: string, version: string): Promise<IUiEventsFrameworkProvider>;
}
export interface IUiEventsFrameworkProvider {
  requestHelper: IRequestHelper;
  dispose: () => void;
  getVersion(): string;
  getApplicationName(): string;
  getCurrentTabId(): Promise<string>;
  getVersions(): IReleaseDetails;
  getActiveTabId(): Promise<string>;
  getAvailableRecords(tabId?: string): Promise<IRecordContext[]>;
  getActiveTab(): Promise<ITabContext>;
  getTabContext(browserTabId?: string, msiTabId?: string): Promise<ITabContext>;
  getCurrentBrowserTabContext(browserTabId?: string): Promise<ITabContext>;
  getAvailableTabs(): Promise<ITabContext[]>;
  getGlobalContext(): Promise<IGlobalContext>;
  getModalWindowContext(): Promise<IModalWindowContext>;
  getSidePaneContext(sidePaneId: string): Promise<ISidePaneContext>;
  getNotificationContext(notificationId: string): Promise<INotificationContext>;
```



```
getMultiChannelAdaptorContext(): Promise<IMultiChannelAdaptorContext>;
getInteractionLogger(channelType: string, contactId?: string): IInteractionLogger;
getInteractionRecorder(channelType: string, contactId?: string): IInteractionRecorder;
getActiveEngagements(): Promise<IEngagementInfo[]>;
}

export interface ISubscriptionContext {
  dispose: () => void;
}

export interface IContext {
  subscribe: (requestObject: IEventRequest, callbackFunction: (response:
  IEventResponse) => void) => ISubscriptionContext;
  subscribeOnce: (requestObject: IEventRequest, callbackFunction: (response:
  IEventResponse) => void) => ISubscriptionContext;
  publish: (requestObject: IOperationRequest) => Promise<IOperationResponse>;
  dispose: () => void;
  getSupportedEvents(): string[];
  getSupportedActions(): string[];
}
export interface IEventRequest {
  getEventName(): string;
}

export interface IInsightsActionRequest {
  setActionDetails(name: string, id: string, payload?: any): void;
  getData(): IActionPayload;
}

export interface IOperationRequest {
  getOperationName(): string;
  setPropagateToTabs(propagateToTabs: boolean): void;
  isPropagateToTabsEnabled(): boolean;
}

export interface IGlobalContext extends IContext {
  getGlobalContextId(): string;
}

export interface ISidePaneContext extends IContext {
  getSidePaneContextId(): string;
}

export interface IInsightsContext extends IContext {
  getContextType(): string;
}
export interface ISmartActionContext extends IContext {
  getContextType(): string;
}

export interface INotificationContext extends IContext {
  getNotificationContextId(): string;
}

export interface ITabContext extends IContext {
  getType(): string;
  getAvailableTabs(): Promise<ITabContext[]>;
  getAvailableRecords(): Promise<IRecordContext[]>;
  getActiveRecord(): Promise<IRecordContext>;
  getDependentTabs(): Promise<ITabContext[]>;
  getNotificationContext(notificationId: string): Promise<INotificationContext>;
  getEngagementInfo(): Promise<IEngagementInfo>;
  getTabInfo(): ITabInfo;
  getModalWindowContext(): Promise<IModalWindowContext>;
}
```

```
export interface IRecordState {
  getView(): string;
  getEditMode(): string;
}

export interface IRecordContext extends IContext {
  getRecordType(): string;
  getRecordId(): string;
  getInsightsContext(): Promise<IIInsightsContext>;
  getSmartActionContext(): Promise<ISmartActionContext>;
  getCurrentView(): Promise<IRecordState>;
}

export interface IRequestHelper {
  createSubscriptionRequest(eventName: string): IEventRequest;
  createPublishRequest(operationName: string): IOperationRequest;
}

export interface IFieldValueChangeEventRequest extends IEventRequest {
  setFields: (fields: string[]) => void;
}

export interface ISetFieldValueOperationRequest extends IOperationRequest {
  field: () => ISetFieldValueRequest;
  disableValidation: () => void;
  disableQueuing: () => void;
}

export interface ISetFieldValueRequest {
  setValue: (fieldName: string, value: any) => void;
}

export interface ISetChannelAvailabilityRequest extends IOperationRequest {
  setAvailability: (isAvailable: boolean) => void;
  setChannel: (channel: string) => void;
}

export interface ISetChannelAvailabilityResponse extends IOperationResponse {
  isSuccess: () => boolean;
  getError: () => any;
}

export interface IGetFieldValueOperationRequest extends IOperationRequest {
  setFields: (fields: string[]) => void;
}

export interface IInsightsSubscriptionRequest extends IEventRequest {
  setId(id: string): void;
}

export interface IWindowSubscriptionRequest extends IEventRequest {
  setId(id: string): void;
}

export interface IContextOpenEventResponse extends IEventResponse {
  getResponseData(): IRecordContext;
}

export interface ITabCloseEventResponse extends IEventResponse {
  getResponseData(): ITabInfo;
  getType(): string;
}

export interface IGetAgentInfoResponse extends IOperationResponse {
  getFirstName(): string;
  getLastName(): string;
  getEmailAddress(): string;
  getUserName(): string;
  getPartyId(): string;
}
```

```
}
export interface ITabCloseOperationResponse extends IOperationResponse {
  getResponseData(): ITabInfo;
  getType(): string;
}
export interface IGetFieldValueResponse extends IOperationResponse {
  getResponseData(): IFieldValueCollection;
}
export interface IGetFieldsInUIResponse extends IOperationResponse {
  getResponseData(): IGetFieldsInUIResponseData;
}
export interface ISetFieldValueResponse extends IOperationResponse {
  getResponseData(): IOperationSuccessData;
}

export interface ISidePaneActionResponse extends IOperationResponse {
  getResponseData(): IOperationSuccessData;
}
export interface IOnAfterSaveEventResponse extends IEventResponse {
  getResponseData(): IOnAfterExtensionContext;
}

export interface ITabEventResponse extends IEventResponse {
  getResponseData(): ITabContext;
  getType(): string;
}

export interface IOnWrapUpTimerExpiredEventResponse extends IEventResponse {
  getResponseData(): IEngagementData;
}

export interface IEngagementData {
  getEngagementId(): string;
  getEngagementType(): string;
}

export interface ITabChangeEventResponse extends IEventResponse {
  getResponseData(): ITabChangeResponse;
  getType(): string;
}

export interface ITabChangeResponse {
  getCurrentTab(): ITabContext;
  getPreviousTab(): ITabContext;
}

export interface ISaveRecordResponse extends IOperationResponse {
  getResponseData(): IOnAfterExtensionContext;
}
export interface IComposeMessageDataResponse extends IOperationResponse {
  getResponseData(): IComposeMessageData;
}
export interface IComposeMessageData {
  getMessageType(): string;
}

export interface IFieldValueChangeEventResponse extends IEventResponse {
  getResponseData(): IFieldValueChangeData;
}
export interface IContextResponse extends IEventResponse {
  getResponseData(): IObjectContext;
}
export interface ISidePaneCloseEventResponse extends IEventResponse {
  getResponseData(): ISidePaneCloseData;
}
export interface IErrorData {
```

```
getStatus(): string;
getMessage(): string;
getErrorDetails(): any;
}

export interface IOperationSuccessData {
  getMessage(): => string;
}

export interface IObjectContext extends ITabInfo {
  getObjectType(): string;
  getObjectId(): string;
}

export interface ITabInfo {
  getTabId(): string;
  getMsiTabId(): string;
  getMsiSubTabId(): string;
}

export interface IEngagementInfo {
  getEngagementId(): string;
  getEngagementType(): string;
  getInteractionId(): string;
  getWrapUpId(): string;
  getInData(): Record<any, any>;
  getOutData(): Record<any, any>;
  isActiveEngagement(): boolean;
}

export interface IReleaseDetails {
  getFusionVersion(): string;
  getUefVersion(): string;
}

export interface IExtensionResponse {
  getContext(): IObjectContext;
}

export interface IEventResponse extends IExtensionResponse {
  getEventName(): string;
}

export interface IOperationResponse extends IExtensionResponse {
  getOperationName(): string;
}

export interface IFieldValueCollection {
  getField: (key: string) => IFieldData;
}

export interface IGetFieldsInUIResponseData {
  getFieldsList(): => string[];
}

export interface IFieldData {
  getValue(): => string;
  getFieldName(): => string;
}

export interface IOnAfterExtensionContext {
  getObjectType(): string;
  getOldObjectId(): string;
  getObjectId(): string;
}

export interface ISidePaneCloseData {
  getId(): string;
}
```

```
export interface IFieldValueChangeData {
  getFieldName: () => string;
  getOldValue: () => string | number | boolean;
  getNewValue: () => string | number | boolean;
}
export interface ISetToolbarPropertiesActionRequest extends IOperationRequest {
  setWidth: (width: Pixels) => void;
  setHeight: (width: Pixels) => void;
}

export interface ICustomEventRequest extends IOperationRequest {
  setCustomEventName(eventName: string): void;
  setEventPayload(payload: any): void;
}
export interface ICustomEventResponse extends IOperationResponse {
  getResponseData(): ICustomEventResponseData;
}
export interface ICustomEventResponseData {
  getData(): any;
  getCustomEventName(): string;
}
export interface IFocusTabResponseData extends IOperationResponse {
  getResponseData(): ITabChangeResponse;
}
export interface IPopFlowUrlRequest {
  setUrl(id: string): void;
}
export interface IPopFlowRequest extends IOperationRequest {
  setOpenPageInNewBrowserTab(value: boolean): void;
  setInputParameters(parameters: any): void;
  setFormView(view: string): void;
}

export interface IPopFlowInAppRequest extends IPopFlowRequest {
  setRecordType(entity: string): void;
  setRecordId(id: string): void;
}
export interface IPopFlowAppUIRequest extends IPopFlowRequest {
  setApplicationUIName(id: string): void;
  setPage(id: string): void;
  setFlow(id: string): void;
  setTabInfo(parameters: any): void;
}

export interface IPopFlowGenericRequest extends IPopFlowRequest {
  setFlow(id: string): void;
  setPage(id: string): void;
}
export interface IPopFlowResponse extends IOperationResponse {
  getResponseData(): ITabContext;
}
export interface IFocusTabResponseData extends IOperationResponse {
  getResponseData(): ITabChangeResponse;
}
export interface IServiceConnectionRequest extends IOperationRequest {
  setServiceConnectionId(url: string): void;
  setParameters(parameter: any): void;
  setBody(body: any): void;
  setApplicationExtensionId(extensionId: string): void;
}

export interface IServiceConnectionResponse extends IOperationResponse {
  getResponseData(): IServiceConnectionResponseData;
}
export interface IServiceConnectionResponseData {
```

```
getBody(): any;
getStatus(): IServiceConnectionStatus;
}
export interface IServiceConnectionStatus {
  getStatusCode(): string;
  getStatusText(): string;
}
export interface IUpdateSidePaneRequest extends IOperationRequest {
  setVisibility: (visibility: boolean) => void;
  setSectionId: (sectionId: string) => void;
  setIcon: (icon: string) => void;
}

export interface IGetGenAiRequest extends IOperationRequest {
  setUseCase: (useCase: string) => void;
  setFieldToUpdate: (fieldName: string) => void;
  setFieldNames: (fieldNames: string[]) => void;
  setConversation: (conversation: string) => void;
}

export interface ISidePaneOpenEventResponse extends IEventResponse {
  getResponseData(): ISidePaneData;
}

export interface INotificationActionEventResponse extends IEventResponse {
  getResponseData(): INotificationActionData;
}

export interface IGetGenAiActionResponse extends IOperationResponse {
  getResponseData(): IGetGenAiActionData;
}
export interface IGetGenAiActionData {
  getSummary(): string;
  getCompletionId(): string;
  getSetFieldValueResponse(): ISetFieldValueResponse | IErrorData;
  getConversationScore(): number;
}

export interface INotificationActionData {
  getActionId(): string;
  getActionName(): string;
  getNotificationId(): string;
}
export interface INotificationCloseActionEventResponse extends IEventResponse {
  getResponseData(): INotificationCloseActionData;
}

export interface INotificationCloseActionData {
  getNotificationId(): string;
  getReason(): string;
}

export interface IWindowCloseActionEventResponse extends IEventResponse {
  getResponseData(): IWindowCloseActionData;
}

export interface IWindowCloseActionData {
  getWindowId(): string;
  getReason(): string;
}

export interface IInsightsActionEventResponse extends IEventResponse {
  getResponseData(): IInsightsActionData;
}

export interface IInsightsActionData {
```

```
getActionId(): string;
getActionName(): string;
getInsightsId(): string;
}

export interface IInsightsDismissActionEventResponse extends IEventResponse {
getResponseData(): IInsightsDismissActionData;
}

export interface IInsightsDismissActionData {
getInsightsId(): string;
getReason(): string;
}

export interface ICustomEventRequest extends IOperationRequest {
setCustomEventName(eventName: string): void;
setEventPayload(payload: any): void;
}

export interface INotificationActionPayload {
id: string;
name: string;
}
export interface IShowNotificationRequest extends IOperationRequest {
setIcon(icon: string): void;
setTitle(title: string): void;
setClosable(isClosable: boolean): void;
setAutoTimeout(time: number): void;
setType(type: string): void;
setSummary(summary: string): void;
setActions(actions: INotificationActionPayload[]): void;
}

export interface IActionPayload {
id: string;
name: string;
payload?: Record<string, any>;
}

export interface IShowInsightsRequest extends IOperationRequest {
action: () => IInsightsActionRequest;
setTitle(title: string): void;
setMessage(summary: string): void;
setHeader(header: string): void;
setId(title: string): void;
setType(type: string): void;
setRank(rank: number): void;
setIcon(icon: string): void;
}

export interface IDismissInsights extends IOperationRequest {
setId(title: string): void;
}

export interface IExecuteSmartAction extends IOperationRequest {
setId(id: string): void;
setParameters(params: Record<string, any>): void;
}

export interface IComposeMessageActionRequest extends IOperationRequest {
setMessageType(messageType: string): void;
setMessageData(): IComposeMessageDataRequest;
}

export interface IComposeMessageDataRequest {
setToAddress(partyName: string, partyId: string, partyViaEndPoint: string, viaId: string,
```

```
recipientPartyTypeCd: string, recipientAlternateId: string): void;
setCCAddress(partyName: string, partyId: string, partyViaEndPoint: string, viaId: string,
recipientPartyTypeCd: string): void;
setBCCAddress(partyName: string, partyId: string, partyViaEndPoint: string, viaId: string,
recipientPartyTypeCd: string): void;
setSubject(subject: string): void;
setMessageBody(messageContent: string): void;
setFieldValue(fieldName: string, fieldValue: string): void;
}

export interface ISmartActionOperationResponse extends IOperationResponse {
getResponseData(): ISmartActionOperationResponseData;
}
export interface ISmartActionOperationResponseData {
getId(): string;
}

export interface IOpenModalWindowRequest extends IOperationRequest {
setURL(url: string): void;
setId(id: string): void;
setTitle(title: string): void;
setClosable(isClosable: boolean): void;
setStyle(style: any): void;
}

export interface IOpenPopupWindowRequest extends IOperationRequest {
setURL(url: string): void;
setId(id: string): void;
setTitle(title: string): void;
setClosable(isClosable: boolean): void;
setStyle(style: any): void;
}
export interface IModalWindowOperationResponse extends IOperationResponse {
getResponseData(): IModalWindowOperationResponseData;
}
export interface IModalWindowOperationResponseData {
getId(): string;
}

export interface IPopupWindowOperationResponse extends IOperationResponse {
getResponseData(): IPopupWindowOperationResponseData;
}
export interface IPopupWindowOperationResponseData {
getId(): any;
}
export interface ICloseModalWindowRequest extends IOperationRequest {
setId(id: string): void;
}

export interface IClosePopupWindowRequest extends IOperationRequest {
setId(id: string): void;
}

export interface INotificationOperationResponse extends IOperationResponse {
getResponseData(): INotificationOperationResponseData;
}
export interface INotificationOperationResponseData {
getNotificationId(): string;
}

export interface IInsightsOperationResponse extends IOperationResponse {
getResponseData(): IInsightsOperationResponseData;
}
export interface IInsightsOperationResponseData {
getInsightsId(): string;
}
```



```
export interface IInsightsGetAllOperationResponse extends IOperationResponse {
  getResponseData(): IInsightsGetAllOperationResponseData;
}

export interface IInsightsGetAllOperationResponseData {
  getInsights(): any;
}

export interface ISidePaneData {
  getActiveSectionId(): string;
  getId(): string;
}

export interface ICustomEventSubscriptionRequest extends IEventRequest {
  setCustomEventName(eventName: string): void;
}

export interface ICustomEventSubscriptionResponse extends IEventResponse {
  getResponseData(): ICustomEventSubscriptionResponseData;
}
export interface ICustomEventSubscriptionResponseData {
  getData(): any;
  getCustomEventName(): string;
}
export interface IDataLoadEventResponse extends IEventResponse {
  getResponseData(): IDataLoadEventData;
}
export interface IDataLoadEventData {
  getCurrentView(): string;
}

export interface IMcaEventResponse extends IEventResponse {
  getResponseData(): IMcaEventResponseData;
}
export interface IMcaEventResponseData {
  getEngagementId(): string;
  getEngagementType(): McaChannels;
  getAssociatedTabContext(): Promise<ITabContext>;
}

export interface IInteractionLogger {
  startLogging(interactionPayload?: Record<string, string>): Promise<string>;
  pauseLogging(): void;
  resumeLogging(): void;
  stopLogging(interactionPayload?: Record<string, string>, wrapUpPayload?: Record<string, string>): void;
}

export interface IInteractionRecorder {
  startRecording(interactionPayload?: Record<string, string>): Promise<string>;
  pauseRecording(): void;
  resumeRecording(): void;
  stopRecording(interactionPayload?: Record<string, string>, wrapUpPayload?: Record<string, string>): void;
}

export interface IMultiChannelAdaptorContext extends IContext {
  getActiveChannels(): Promise<string[]>;
  getCommunicationChannelContext(channelType: McaChannels): Promise<ICommunicationChannelContext>;
}

export interface ICommunicationChannelContext extends IContext {
  getChannelType(): McaChannelTypes;
  getChannel(): McaChannels;
}

export interface IPhoneContext extends ICommunicationChannelContext {
```

```
getChannelType(): McaChannelTypes;
getChannel(): McaChannels;
getFrameOrigin(): string;
getToolbarName(): string;
getVersion(): string;
getEventSource(): string;
getToolbarContext(): IToolbarContext;
}

export interface IChatContext extends ICommunicationChannelContext {
getChannelType(): McaChannelTypes;
getChannel(): McaChannels;
}

export interface IEngagementContext extends IContext {
getChannelType(): McaChannelTypes;
getChannel(): McaChannels;
getEngagementId(): string;
getTabContext(): Promise<ITabContext>;
getEngagementData(): IMcaEngagementData;
getWrapupContext(): Promise<IRecordContext>;
}

export interface IMcaActionRequest extends IOperationRequest {
setAppClassification: (appClassification: string) => void;
setEventId: (eventId: string) => void;
}

export interface IMcaNewCommEventActionRequest extends IMcaActionRequest {
setLookupObject: (lookupObject: string) => void;
setInputData: (inData: IMcaInData) => void;
getInData: () => IMcaInDataRequest;
}

export interface IMcaStartCommEventActionRequest extends IMcaActionRequest {
setInputData: (inData: IMcaStartCommInData) => void;
getInData: () => IMcaStartCommInDataRequest;
}

export interface IMcaCloseCommEventActionRequest extends IMcaActionRequest {
setReason: (reason: string) => void;
setInputData: (inData: IMcaCloseCommInData) => void;
getInData: () => IMcaInDataRequest;
}

export interface IMcaFeedLiveTranscriptActionRequest extends IMcaActionRequest {
setMessage: (message: string) => void;
setMessageId: (messageId: string) => void;
setRole: (role: string) => void;
setState: (state: string) => void;
setTime: (time: number) => void;
setUsername: (username: string) => void;
setUserId: (userId: string) => void;
}

export interface IMcaFeedSystemMessageActionRequest extends IMcaActionRequest {
setType: (type: string) => void;
setIconName: (icon: string) => void;
setMessageId: (messageId: string) => void;
setMessage: (message: string) => void;
setUsername: (username: string) => void;
setUserId: (userId: string) => void;
setTime: (time: number) => void;
}

export interface IMcaFeedOfflineTranscriptActionRequest extends IMcaActionRequest {
createAgentTranscript: () => ITranscriptItem;
createEndUserTranscript: () => ITranscriptItem;
addTranscript: (transcriptItem: ITranscriptItem) => void;
}

export interface ITranscriptItem {
setMessage: (message: string) => void;
setMessageId: (messageId: string) => void;
```

```
    setTime: (time: number) => void;
    setUsername: (username: string) => void;
    setUserId: (userId: string) => void;
  }
  export interface IMcaNewComActionResponse extends IOperationResponse {
    getResponseData(): IMcaNewComActionData;
  }
  export interface IMcaStartComActionResponse extends IOperationResponse {
    getResponseData(): IMcaStartComActionData;
  }

  export interface IMcaCloseComActionResponse extends IOperationResponse {
    getResponseData(): IMcaCloseComActionData;
  }
  export interface ISetToolbarPropertiesActionResponse extends IOperationResponse {
    getResponseData(): ISetToolbarPropertiesActionResponseData;
    isSuccess(): boolean;
  }

  export interface IMcaNewComActionData {
    getData(): IMcaNewComEventActionData;
    getOutputData(): IMcaOutDataResponse;
    getOutData(): IMcaOutData;
  }
  export interface IMcaStartComActionData {
    getData(): IMcaStartComEventActionData;
    getOutData(): IMcaStartCommEventOutData;
    getEngagementContext(): IEngagementContext;
    getOutputData(): IMcaStartCommEventOutDataResponse;
  }
  export interface IMcaCloseComActionData {
    getData(): IMcaCloseComEventActionData;
    getOutData(): IMcaCloseCommEventOutData;
    getEngagementContext(): IEngagementContext;
    getOutputData(): IMcaCloseCommEventOutDataResponse;
  }
  export interface ISetToolbarPropertiesActionResponseData {
    status: string;
    payload: any;
  }
  export interface IMcaEngagementData {
    getData(): IMcaStartComEventActionData;
    getOutData(): IMcaStartCommEventOutData;
    getOutputData(): IMcaStartCommEventOutDataResponse;
    getEngagementId(): string;
    getWrapUpId(): string;
  }
  export interface IMcaOutData {
    SVC_MCA_EMAIL: string;
    SVC_MCA_COMMUNICATION_DIRECTION: string;
    SVC_MCA_CONTACT_NAME: string;
    channel: string;
    channelType: string;
    SVC_MCA_DISPLAY_NAME: string;
    notificationType: string;
    lookupObject: string;
    SVC_MCA_OFFER_TIMEOUT_SEC: string;
    SVC_MCA_CONTACT_FIRST_NAME: string;
    SVC_MCA_SR_ID: string;
    appClassification: string;
    callStatus: string;
    channelId: string;
    eventId: string;
    SVC_MCA_CONTACT_PRIMARY_PHONE: string;
    SVC_MCA_INTERACTION_REF_OBJ_TYPE: string;
    SVC_MCA_SR_NUM: string;
  }
```

```
SVCMCA_WRAPUP_TIMEOUT: string;
SVCMCA_CONTACT_LAST_NAME: string;
phoneLineId: string;
SVCMCA_CONTACT_NUMBER: string;
callDrection: string;
SVCMCA_CONTACT_ID: string;
SVCMCA_ANI: string;
SVCMCA_CONTACT_PRIM_ORG_NAME: string;
SVCMCA_INTERACTION_REF_OBJ_ID: string;
SVCMCA_CONTACT_ORG_ID: string;
SVCMCA_INTERACTION_ID: string;
}

export interface IMcaStartCommEventOutData extends IMcaOutData {
SVCMCA_CHANNEL_ID: string;
SVCMCA_UI_TYPE_CD: string;
SVCMCA_SR_TITLE: string;
SVCMCA_WRAPUP_ID: string;
}

export interface IMcaCloseCommEventOutData extends IMcaOutData {
SVCMCA_CHANNEL_ID: string;
SVCMCA_UI_TYPE_CD: string;
SVCMCA_SR_TITLE: string;
SVCMCA_WRAPUP_ID: string;
WrapupStartTime: string;
}

export interface IMcaInData {
eventId: string;
callStatus: string;
callDirection: string;
appClassification: string;
notificationType: string;
lookupObject: string;
callSource: string;
callDestination: string;
channelType: string;
comPanelMsg1: string;
comPanelMsg2: string;
comPanelQName1: string;
comPanelQName2: string;
comPanelQName3: string;
averageWaitTime: string;
numberOfCalls: string;
agentMsg2ComPanel: string;
phoneLineId: string;
chatTestMode: string;
SVCMCA_ANI?: string;
SVCMCA_OFFER_TIMEOUT_SEC?: string;
channel: string;
SVCMCA_SR_NUM?: string;
SVCMCA_COMMUNICATION_DIRECTION: string;
SVCMCA_WRAPUP_TIMEOUT?: string;
}

export interface IMcaStartCommInData extends IMcaInData {
SVCMCA_EMAIL: string;
SVCMCA_CONTACT_NAME: string;
channel: string;
SVCMCA_DISPLAY_NAME: string;
SVCMCA_CONTACT_FIRST_NAME: string;
SVCMCA_SR_ID: string;
channelId: string;
SVCMCA_CONTACT_PRIMARY_PHONE: string;
SVCMCA_INTERACTION_REF_OBJ_TYPE: string;
SVCMCA_CONTACT_LAST_NAME: string;
SVCMCA_CONTACT_NUMBER: string;
}
```

```
SVCMCA_CONTACT_ID: string;
SVCMCA_CONTACT_PRIM_ORG_NAME: string;
SVCMCA_INTERACTION_REF_OBJ_ID: string;
SVCMCA_CONTACT_ORG_ID: string;
SVCMCA_INTERACTION_ID: string;
SVCMCA_2NDARY_CALL_TYPE?: string;
SVCMCA_INITIATING_AGENT_NAME?: string;
SVCMCA_INITIATING_AGENT_ID?: string;
SVCMCA_AGENT_NUMBER?: string;
SVCMCA_PARENT_INTERACTION_ID?: string;
}
export interface IMcaCloseCommInData extends IMcaInData {
SVCMCA_EMAIL: string;
SVCMCA_CONTACT_NAME: string;
channel: string;
SVCMCA_DISPLAY_NAME: string;
SVCMCA_CONTACT_FIRST_NAME: string;
SVCMCA_SR_ID: string;
channelId: string;
SVCMCA_CONTACT_PRIMARY_PHONE: string;
SVCMCA_INTERACTION_REF_OBJ_TYPE: string;
SVCMCA_CONTACT_LAST_NAME: string;
SVCMCA_CONTACT_NUMBER: string;
SVCMCA_CONTACT_ID: string;
SVCMCA_CONTACT_PRIM_ORG_NAME: string;
SVCMCA_INTERACTION_REF_OBJ_ID: string;
SVCMCA_CONTACT_ORG_ID: string;
SVCMCA_INTERACTION_ID: string;
SVCMCA_CHANNEL_ID: string;
SVCMCA_UI_TYPE_CD: string;
SVCMCA_SR_TITLE: string;
SVCMCA_WRAPUP_ID: string;
}
export interface IMcaInDataRequest {
setCallDestination: (value: string) => void;
setCallDirection: (value: string) => void;
setAppClassification: (value: string) => void;
setNotificationType: (value: string) => void;
setLookupObject: (value: string) => void;
setCallSource: (value: string) => void;
setCallStatus: (value: string) => void;
setChannelType: (value: string) => void;
setComPanelMsg1: (value: string) => void;
setComPanelMsg2: (value: string) => void;
setComPanelQName1: (value: string) => void;
setComPanelQName2: (value: string) => void;
setComPanelQName3: (value: string) => void;
setAverageWaitTime: (value: string) => void;
setNumberOfCalls: (value: string) => void;
setAgentMsg2ComPanel: (value: string) => void;
setPhoneLineId: (value: string) => void;
setChatTestMode: (value: string) => void;
setSVCMCA_ANI: (value: string) => void;
setSVCMCA_WRAPUP_TIMEOUT: (value: string) => void;
setSVCMCA_OFFER_TIMEOUT_SEC: (value: string) => void;
setSVCMCA_COMMUNICATION_DIRECTION: (value: string) => void;
setSVCMCA_SR_NUM: (value: string) => void;
setChannel: (value: string) => void;
setInDataValueByAttribute: (attribute: string, value: any) => void;
}
export interface IMcaStartCommInDataRequest extends IMcaInDataRequest {
setSVCMCA_EMAIL: (value: string) => void;
setSVCMCA_CONTACT_NAME: (value: string) => void;
setChannel: (value: string) => void;
setSVCMCA_DISPLAY_NAME: (value: string) => void;
setSVCMCA_CONTACT_FIRST_NAME: (value: string) => void;
```

```
    setSVCMA_SR_ID: (value: string) => void;
    setChannelId: (value: string) => void;
    setSVCMA_CONTACT_PRIMARY_PHONE: (value: string) => void;
    setSVCMA_INTERACTION_REF_OBJ_TYPE: (value: string) => void;
    setSVCMA_CONTACT_LAST_NAME: (value: string) => void;
    setSVCMA_CONTACT_NUMBER: (value: string) => void;
    setSVCMA_CONTACT_ID: (value: string) => void;
    setSVCMA_CONTACT_PRIM_ORG_NAME: (value: string) => void;
    setSVCMA_INTERACTION_REF_OBJ_ID: (value: string) => void;
    setSVCMA_CONTACT_ORG_ID: (value: string) => void;
    setSVCMA_INTERACTION_ID: (value: string) => void;
  }

  export interface IMcaCloseCommInDataRequest extends IMcaInDataRequest {
    setSVCMA_EMAIL: (value: string) => void;
    setSVCMA_CONTACT_NAME: (value: string) => void;
    setChannel: (value: string) => void;
    setSVCMA_DISPLAY_NAME: (value: string) => void;
    setSVCMA_CONTACT_FIRST_NAME: (value: string) => void;
    setSVCMA_SR_ID: (value: string) => void;
    setChannelId: (value: string) => void;
    setSVCMA_CONTACT_PRIMARY_PHONE: (value: string) => void;
    setSVCMA_INTERACTION_REF_OBJ_TYPE: (value: string) => void;
    setSVCMA_CONTACT_LAST_NAME: (value: string) => void;
    setSVCMA_CONTACT_NUMBER: (value: string) => void;
    setSVCMA_CONTACT_ID: (value: string) => void;
    setSVCMA_CONTACT_PRIM_ORG_NAME: (value: string) => void;
    setSVCMA_INTERACTION_REF_OBJ_ID: (value: string) => void;
    setSVCMA_CONTACT_ORG_ID: (value: string) => void;
    setSVCMA_INTERACTION_ID: (value: string) => void;
    setSVCMA_UI_TYPE_CD: (value: string) => void;
    setSVCMA_CHANNEL_ID: (value: string) => void;
    setSVCMA_SR_TITLE: (value: string) => void;
    setSVCMA_WRAPUP_ID: (value: string) => void;
  }

  export interface IMcaComEventActionData {
    channelId: string;
    channelType: string;
    engagementId: string;
    eventId: string;
    eventSource: string;
    interactionId: string;
    method: string;
    result: string;
    toolbarName: string;
    uuid: string;
    channel: McaChannels;
  }

  export interface IMcaNewComEventActionData extends IMcaComEventActionData {
    outData: IMcaOutData;
  }

  export interface IMcaStartComEventActionData extends IMcaComEventActionData {
    outData: IMcaStartCommEventOutData;
    screenPopMode: string;
    wrapupId: string;
  }

  export interface IMcaCloseComEventActionData extends IMcaComEventActionData {
    outData: IMcaCloseCommEventOutData;
    wrapupId: string;
  }

  export interface IMcaOutDataResponse {
    getSVCMA_COMMUNICATION_DIRECTION: () => string;
    getSVCMA_CONTACT_NAME: () => string;
    getSVCMA_EMAIL: () => string;
```

```
getChannel: () => string;
getChannelType: () => string;
getSVCMA_DISPLAY_NAME: () => string;
getNotificationType: () => string;
getLookupObject: () => string;
getSVCMA_OFFER_TIMEOUT_SEC: () => string;
getSVCMA_CONTACT_FIRST_NAME: () => string;
getSVCMA_SR_ID: () => string;
getAppClassification: () => string;
getCallStatus: () => string;
getChannelId: () => string;
getEventId: () => string;
getSVCMA_CONTACT_PRIMARY_PHONE: () => string;
getSVCMA_INTERACTION_REF_OBJ_TYPE: () => string;
getSVCMA_SR_NUM: () => string;
getSVCMA_WRAPUP_TIMEOUT: () => string;
getSVCMA_CONTACT_LAST_NAME: () => string;
getPhoneLineId: () => string;
getSVCMA_CONTACT_NUMBER: () => string;
getCallDirection: () => string;
getSVCMA_CONTACT_ID: () => string;
getSVCMA_ANI: () => string;
getSVCMA_CONTACT_PRIM_ORG_NAME: () => string;
getSVCMA_INTERACTION_REF_OBJ_ID: () => string;
getSVCMA_CONTACT_ORG_ID: () => string;
getSVCMA_INTERACTION_ID: () => string;
getValueByAttribute: (attribute: string) => string;
}

export interface IMcaCloseCommEventOutDataResponse extends IMcaOutDataResponse {
getSVCMA_CHANNEL_ID: () => string;
getSVCMA_UI_TYPE_CD: () => string;
getSVCMA_SR_TITLE: () => string;
getSVCMA_WRAPUP_ID: () => string;
getWrapupStartTime: () => string;
}

export interface IMcaStartCommEventOutDataResponse extends IMcaOutDataResponse {
getSVCMA_CHANNEL_ID: () => string;
getSVCMA_UI_TYPE_CD: () => string;
getSVCMA_SR_TITLE: () => string;
getSVCMA_WRAPUP_ID: () => string;
}

type McaChannelTypes = string;
type McaChannels = string;

export interface IMcaGetConfigurationActionRequest extends IOperationRequest {
setConfigType: (configType: IMCAGetConfigurationConfigTypes) => void;
}

export interface IMcaAgentStateEventActionRequest extends IOperationRequest {
setEventId: (eventId: string) => void;
setIsAvailable: (isAvailable: boolean) => void;
setIsLoggedIn: (isLoggedIn: boolean) => void;
setState: (stateCd: string) => void;
setStateDisplayString: (stateDisplayString: string) => void;
setReason: (reasonCd: string) => void;
setReasonDisplayString: (reasonDisplayString: string) => void;
setInData: (inData: Record<string, any>) => void;
setCtiConfig: (inData: Record<string, any>) => void;
disableEngagementPanelForOutbound: () => void;
}

export interface IMcaDisableFeatureActionRequest extends IOperationRequest {
setFeatures: (features: string[]) => void;
}
```

```
export interface IMcaReadyForOperationActionRequest extends IOperationRequest {
  setReadiness: (readiness: boolean) => void;
}

export interface IMcaGetConfigurationActionResponse extends IOperationResponse {
  getResponseData(): IMcaGetConfigurationActionResponseData;
}

export interface IMcaDisableFeatureActionResponse extends IOperationResponse {
  getResponseData(): IMcaDisableFeatureActionResponseData;
}

export interface IMcaReadyForOperationActionResponse extends IOperationResponse {
  getResponseData(): IMcaReadyForOperationActionResponseData;
}

export interface IMcaAgentStateEventActionResponse extends IOperationResponse {
  getResponseData(): IMcaAgentStateEventActionResponseData;
}

export interface IMcaOutboundCommErrorActionRequest extends IOperationRequest {
  setCommUuid(commUuid: string): void;
  setErrorCode(errorCode: string): void;
  setErrorMsg(errorMsg: string): void;
}

export interface IMcaOutBoundCommErrorActionResponse extends IOperationResponse {
  getResponseData(): IMcaOutBoundCommErrorActionResponseData;
}

export interface IMcaGetConfigurationActionResponseData {
  getData(): IMCAGetConfigurationResponsePayload;
  getConfiguration(): IMCAGetConfiguration;
  isSuccess(): boolean;
  getError(): any;
}

export interface IMcaOutBoundCommErrorActionResponseData {
  getData(): IMCAOutBoundCommErrorResponsePayload;
  isSuccess(): boolean;
  getError(): string;
}

export interface IMcaAgentStateEventActionResponseData {
  getData(): any;
  isSuccess(): boolean;
  getError(): any;
}

export interface IMcaDisableFeatureActionResponseData {
  getData(): any;
}

export interface IMcaReadyForOperationActionResponseData {
  getData(): any;
}

export interface IMCAGetConfigurationResponsePayload {
  outData: Record<string, any>;
  method: any;
  configuration: IMCAConfiguration;
  result: string;
  toolbarName: string;
  uuid: string;
  error: any;
}
```



```
}

export interface IMCAOutBoundCommErrorResponsePayload {
  result: string;
  error: string;
}

export interface IMCAAgentStateEventResponsePayload {
  method: string;
  result: string;
  toolbarName: string;
  uuid: string;
  error: any;
}

export interface IMCAConfiguration {
  faTrustToken: string;
  agentId: string;
  agentPartyId: string;
  features: string;
  companionPanelUrl: string;
  companionPanelTitle: string;
  userName?: string;
  userFullName?: string;
  userId?: string;
}

export interface IMCAGetConfiguration {
  getAgentId(): string;
  getAgentPartyId(): string;
  getFeatures(): string[];
  getCompanionPanelUrl(): string;
  getCompanionPanelTitle(): string;
  getFaTrustToken(): string;
  getUsername(): string;
  getUserId(): string;
  getUserFullName(): string;
}

export interface IMcaEventRequest extends IEventRequest {
  setAppClassification(appClassification: string): void;
}

export interface IMcaOnDataUpdatedEventResponse extends IEventResponse {
  getResponseData(): IMcaOnDataUpdatedData;
}

export interface IMcaOnDataUpdatedData {
  getData: () => IMcaOnDataUpdatedEventData;
  getOutData: () => IMcaOnDataUpdatedOutData;
  getEventId: () => string;
  getOutputData: () => IMcaOnDataUpdatedOutDataResponse;
}

export interface IMcaOnDataUpdatedOutDataResponse {
  getUpdateType: () => string;
}

export interface IMcaOnDataUpdatedOutData {
  updateType: string;
}

export interface IMcaOnDataUpdatedEventData extends IMcaEventData {
  outData: IMcaOnDataUpdatedOutData;
  eventId: string;
  eventSource: string;
  result: string;
}
```

```
export interface IMcaEventData {
  channel: string;
  channelType: string;
  method: string;
  toolbarName: string;
  uuid: string;
}

export interface IMcaonOutgoingEventResponse extends IEventResponse {
  getResponseData(): IMcaOnOutgoingEventData;
}

export interface IMcaOnOutgoingEventData {
  getData(): => IMcaOnOutgoingEventPayloadData;
  getOutData(): => IMcaOnOutgoingEventOutData;
  getOutputData(): => IMcaOnOutgoingEventOutDataResponse;
}

export interface IMcaOnOutgoingEventPayloadData extends IMcaEventData {
  outData: IMcaOnOutgoingEventOutData;
  origin: string;
  windowName: string;
}

export interface IMcaOnOutgoingEventOutData {
  SVC_MCA_ANI: string;
  SVC_MCA_DISPLAY_NAME: string;
  SVC_MCA_COMMUNICATION_DIRECTION: string;
  SVC_MCA_CALL_ID: string;
  SVC_MCA_CONTACT_ID: string;
  SVC_MCA_CONTACT_JOB_TITLE: string;
  SVC_MCA_INTERACTION_REF_OBJ_ID: string;
  SVC_MCA_INTERACTION_REF_OBJ_TYPE: string;
  SVC_MCA_ORG_ID: string;
}

export interface IMcaOnOutgoingEventOutDataResponse {
  getSVC_MCA_ANI(): => string;
  getSVC_MCA_DISPLAY_NAME(): => string;
  getSVC_MCA_COMMUNICATION_DIRECTION(): => string;
  getSVC_MCA_CALL_ID(): => string;
  getSVC_MCA_CONTACT_ID(): => string;
  getSVC_MCA_CONTACT_JOB_TITLE(): => string;
  getSVC_MCA_INTERACTION_REF_OBJ_ID(): => string;
  getSVC_MCA_INTERACTION_REF_OBJ_TYPE(): => string;
  getSVC_MCA_ORG_ID(): => string;
}

export interface IMcaOnToolbarInteractionCommandEventResponse extends IEventResponse {
  getResponseData(): IMcaOnToolbarInteractionCommandDataResponse;
}

export interface IMcaOnToolbarAgentCommandEventResponse extends IEventResponse {
  getResponseData(): IMcaOnToolbarAgentCommandDataResponse;
}

export interface IMcaOnToolbarInteractionCommandDataResponse {
  getData(): IMcaOnToolbarInteractionCommandData;
  getEventId(): string;
  getCommand(): string;
  getSlot(): string;
  getInData(): any;
  getResult(): string;
  getResultDisplayString(): string;
  getOutdata(): any;
  setResult(value: string): void;
}

export interface IMcaOnToolbarAgentCommandDataResponse {
```

```
getData(): IMcaOnToolbarAgentCommandData;
getEventId(): string;
getCommand(): string;
getInData(): any;
getResult(): string;
getResultDisplayString(): string;
getOutdata(): any;
setOutdata(data: any): void;
setResult(value: string): void;
}

export interface IMcaOnToolbarInteractionCommandData extends IMcaEventData {
command: string;
commandId: string;
eventId: string;
inData: any;
outData: any;
result: string;
target: string;
timeStamp: number;
slot: string;
resultDisplayString: string;
}

export interface IMcaOnToolbarAgentCommandData extends IMcaEventData {
command: string;
eventId: string;
inData: any;
outData: any;
result: string;
timeStamp: number;
resultDisplayString: string;
}

export interface ISmartActionEventRequest extends IEventRequest {
setSmartActionId: (smartActionId: string) => void;
}

export interface ISmartActionEventResponse extends IEventResponse {
getResponseData(): ISmartActionEventData;
}

export interface ISmartActionEventData {
getSmartActionId(): string;
}

export type IMCAGetConfigurationConfigTypes = string;
export type Pixels = number;

export interface IComposeMessageEventRequest extends IEventRequest {
setMessageType: (messageType: string) => void;
getMessageType: () => string;
}
export interface IComposeMessageEventResponse extends IEventResponse {
getResponseData(): IComposeMessageEventData;
}
export interface IComposeMessageEventData {
getMessageType(): string;
}

export interface ISpecialFieldMapEntry {
serviceConnectionEndpoint: string;
fieldReference: string;
specialField: string;
specialFieldReference: string;
isPathComponent: boolean;
```

```

}

}

declare type IUiEventsFramework = CX_SVC_UI_EVENTS_FRAMEWORK.IUiEventsFramework;
declare type IUiEventsFrameworkProvider = CX_SVC_UI_EVENTS_FRAMEWORK.IUiEventsFrameworkProvider;
declare type ISubscriptionContext = CX_SVC_UI_EVENTS_FRAMEWORK.ISubscriptionContext;
declare type IContext = CX_SVC_UI_EVENTS_FRAMEWORK.IContext;
declare type IModalWindowContext = CX_SVC_UI_EVENTS_FRAMEWORK.IModalWindowContext;
declare type IToolBarContext = CX_SVC_UI_EVENTS_FRAMEWORK.IToolBarContext;
declare type IEventRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IEventRequest;
declare type IOperationRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IOperationRequest;
declare type IInsightsActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsActionRequest;
declare type IGlobalContext = CX_SVC_UI_EVENTS_FRAMEWORK.IGlobalContext;
declare type ITabContext = CX_SVC_UI_EVENTS_FRAMEWORK.ITabContext;
declare type IRecordContext = CX_SVC_UI_EVENTS_FRAMEWORK.IRecordContext;
declare type IRequestHelper = CX_SVC_UI_EVENTS_FRAMEWORK.IRequestHelper;
declare type IFieldValueChangeEventRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IFieldValueChangeEventRequest;
declare type IFieldValueChangeEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IFieldValueChangeEventResponse;
declare type ISetFieldValueOperationRequest = CX_SVC_UI_EVENTS_FRAMEWORK.ISetFieldValueOperationRequest;
declare type ISetFieldValueRequest = CX_SVC_UI_EVENTS_FRAMEWORK.ISetFieldValueRequest;
declare type IGetFieldValueOperationRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IGetFieldValueOperationRequest;
declare type IContextOpenEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IContextOpenEventResponse;
declare type ITabEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ITabEventResponse;
declare type ITabCloseEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ITabCloseEventResponse;
declare type IContextResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IContextResponse;
declare type ITabCloseOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ITabCloseOperationResponse;
declare type IGetFieldValueResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IGetFieldValueResponse;
declare type ISetFieldValueResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ISetFieldValueResponse;
declare type IGetFieldsInUIResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IGetFieldsInUIResponse;
declare type IOnAfterSaveEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IOnAfterSaveEventResponse;
declare type ITabChangeEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ITabChangeEventResponse;
declare type ITabChangeResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ITabChangeResponse;
declare type ISaveRecordResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ISaveRecordResponse;
declare type IErrorData = CX_SVC_UI_EVENTS_FRAMEWORK.IErrorData;
declare type IOperationSuccessData = CX_SVC_UI_EVENTS_FRAMEWORK.IOperationSuccessData;
declare type IObjectContext = CX_SVC_UI_EVENTS_FRAMEWORK.IObjectContext;
declare type ITabInfo = CX_SVC_UI_EVENTS_FRAMEWORK.ITabInfo;
declare type IExtensionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IExtensionResponse;
declare type IEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IEventResponse;
declare type IOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IOperationResponse;
declare type IFieldValueCollection = CX_SVC_UI_EVENTS_FRAMEWORK.IFieldValueCollection;
declare type IGetFieldsInUIResponseData = CX_SVC_UI_EVENTS_FRAMEWORK.IGetFieldsInUIResponseData;
declare type IFieldData = CX_SVC_UI_EVENTS_FRAMEWORK.IFieldData;
declare type IOnAfterExtensionContext = CX_SVC_UI_EVENTS_FRAMEWORK.IOnAfterExtensionContext;
declare type IFieldValueChangeData = CX_SVC_UI_EVENTS_FRAMEWORK.IFieldValueChangeData;
declare type ICustomEventRequest = CX_SVC_UI_EVENTS_FRAMEWORK.ICustomEventRequest;
declare type ICustomEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ICustomEventResponse;
declare type ICustomEventResponseData = CX_SVC_UI_EVENTS_FRAMEWORK.ICustomEventResponseData;
declare type ICustomEventSubscriptionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.ICustomEventSubscriptionRequest;
declare type ICustomEventSubscriptionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ICustomEventSubscriptionResponse;
declare type ICustomEventSubscriptionResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.ICustomEventSubscriptionResponseData;
declare type IFocusTabResponseData = CX_SVC_UI_EVENTS_FRAMEWORK.IFocusTabResponseData;
declare type IServiceConnectionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IServiceConnectionRequest;
declare type IServiceConnectionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IServiceConnectionResponse;
declare type IServiceConnectionResponseData = CX_SVC_UI_EVENTS_FRAMEWORK.IServiceConnectionResponseData;
declare type IServiceConnectionStatus = CX_SVC_UI_EVENTS_FRAMEWORK.IServiceConnectionStatus;
declare type IOpenModalWindowRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IOpenModalWindowRequest;
declare type IOpenPopupWindowRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IOpenPopupWindowRequest;
declare type IModalWindowOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IModalWindowOperationResponse;
declare type IModalWindowOperationResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IModalWindowOperationResponseData;
declare type IPopupWindowOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IPopupWindowOperationResponse;
declare type IPopupWindowOperationResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IPopupWindowOperationResponseData;

```

```

declare type ICloseModalWindowRequest = CX_SVC_UI_EVENTS_FRAMEWORK.ICloseModalWindowRequest;
declare type IClosePopupWindowRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IClosePopupWindowRequest;
declare type IUpdateSidePaneRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IUpdateSidePaneRequest;
declare type ISidePaneOpenEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ISidePaneOpenEventResponse;
declare type ISidePaneData = CX_SVC_UI_EVENTS_FRAMEWORK.ISidePaneData;
declare type ISidePaneContext = CX_SVC_UI_EVENTS_FRAMEWORK.ISidePaneContext;
declare type ISidePaneCloseEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ISidePaneCloseEventResponse;
declare type ISidePaneCloseData = CX_SVC_UI_EVENTS_FRAMEWORK.ISidePaneCloseData;
declare type IPopFlowRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IPopFlowRequest;
declare type IPopFlowInAppRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IPopFlowInAppRequest;
declare type IPopFlowGenericRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IPopFlowGenericRequest;
declare type IPopFlowAppUIRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IPopFlowAppUIRequest;
declare type IPopFlowResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IPopFlowResponse;
declare type IGetAgentInfoResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IGetAgentInfoResponse;
declare type IDataLoadEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IDataLoadEventResponse;
declare type IDataLoadEventData = CX_SVC_UI_EVENTS_FRAMEWORK.IDataLoadEventData;
declare type INotificationContext = CX_SVC_UI_EVENTS_FRAMEWORK.INotificationContext;
declare type IInsightsContext = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsContext;
declare type IShowInsightsRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IShowInsightsRequest;
declare type IActionPayload = CX_SVC_UI_EVENTS_FRAMEWORK.IActionPayload;
declare type IDismissInsights = CX_SVC_UI_EVENTS_FRAMEWORK.IDismissInsights;
declare type IInsightsActionEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsActionEventResponse;
declare type IInsightsActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsActionData;
declare type IInsightsDismissActionEventResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsDismissActionEventResponse;
declare type IInsightsDismissActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsDismissActionData;
declare type IInsightsOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsOperationResponse;
declare type IInsightsOperationResponseData = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsOperationResponseData;
declare type IInsightsGetAllOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsGetAllOperationResponse;
declare type IInsightsGetAllOperationResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsGetAllOperationResponseData;
declare type IInsightsSubscriptionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IInsightsSubscriptionRequest;
declare type IWindowSubscriptionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IWindowSubscriptionRequest;
declare type IWindowCloseActionEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IWindowCloseActionEventResponse;
declare type IWindowCloseActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IWindowCloseActionData;
declare type INotificationActionEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.INotificationActionEventResponse;
declare type INotificationActionData = CX_SVC_UI_EVENTS_FRAMEWORK.INotificationActionData;
declare type INotificationCloseActionEventResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.INotificationCloseActionEventResponse;
declare type INotificationCloseActionData = CX_SVC_UI_EVENTS_FRAMEWORK.INotificationCloseActionData;
declare type INotificationActionPayload = CX_SVC_UI_EVENTS_FRAMEWORK.INotificationActionPayload;
declare type IShowNotificationRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IShowNotificationRequest;
declare type INotificationOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.INotificationOperationResponse;
declare type INotificationOperationResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.INotificationOperationResponseData;
declare type IMultiChannelAdaptorContext = CX_SVC_UI_EVENTS_FRAMEWORK.IMultiChannelAdaptorContext;
declare type ICommunicationChannelContext = CX_SVC_UI_EVENTS_FRAMEWORK.ICommunicationChannelContext;
declare type IPhoneContext = CX_SVC_UI_EVENTS_FRAMEWORK.IPhoneContext;
declare type IEngagementContext = CX_SVC_UI_EVENTS_FRAMEWORK.IEngagementContext;
declare type IMcaActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaActionRequest;
declare type IMcaNewCommEventActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaNewCommEventActionRequest;
declare type IMcaStartCommEventActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartCommEventActionRequest;
declare type IMcaCloseCommEventActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseCommEventActionRequest;
declare type ITranscriptItem = CX_SVC_UI_EVENTS_FRAMEWORK.ITranscriptItem;
declare type IMcaFeedLiveTranscriptActionRequest =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaFeedLiveTranscriptActionRequest;
declare type IMcaFeedOfflineTranscriptActionRequest =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaFeedOfflineTranscriptActionRequest;
declare type IMcaFeedSystemMessageActionRequest =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaFeedSystemMessageActionRequest;
declare type IMcaNewComActionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaNewComActionResponse;
declare type IMcaStartComActionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartComActionResponse;
declare type IMcaCloseComActionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseComActionResponse;
declare type IMcaNewComActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaNewComActionData;
declare type IMcaStartComActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartComActionData;
declare type IMcaCloseComActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseComActionData;

```

```

declare type IMcaOutData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOutData;
declare type IMcaStartCommEventOutData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartCommEventOutData;
declare type IMcaCloseCommEventOutData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseCommEventOutData;
declare type IMcaInData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaInData;
declare type IMcaInDataRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaInDataRequest;
declare type IMcaComEventActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaComEventActionData;
declare type IMcaNewComEventActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaNewComEventActionData;
declare type IMcaStartComEventActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartComEventActionData;
declare type IMcaCloseComEventActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseComEventActionData;
declare type IMcaEngagementData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaEngagementData;
declare type IMcaOutDataResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOutDataResponse;
declare type IMcaCloseCommEventOutDataResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseCommEventOutDataResponse;
declare type IMcaStartCommEventOutDataResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartCommEventOutDataResponse;
declare type IMcaCloseCommInDataRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseCommInDataRequest;
declare type IMcaStartCommInDataRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartCommInDataRequest;
declare type IMcaStartCommInData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaStartCommInData;
declare type IMcaCloseCommInData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaCloseCommInData;
declare type IMcaGetConfigurationActionResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaGetConfigurationActionResponse;
declare type IMcaGetConfigurationActionResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaGetConfigurationActionResponseData;
declare type IMcaDisableFeatureActionResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaDisableFeatureActionResponseData;
declare type IMcaDisableFeatureActionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaDisableFeatureActionResponse;
declare type IMCAGetConfigurationResponsePayload =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMCAGetConfigurationResponsePayload;
declare type IMcaGetConfigurationActionRequest =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaGetConfigurationActionRequest;
declare type IMCAGetConfigurationConfigTypes = CX_SVC_UI_EVENTS_FRAMEWORK.IMCAGetConfigurationConfigTypes;
declare type IMCAConfiguration = CX_SVC_UI_EVENTS_FRAMEWORK.IMCAConfiguration;
declare type IMCAGetConfiguration = CX_SVC_UI_EVENTS_FRAMEWORK.IMCAGetConfiguration;
declare type IMcaDisableFeatureActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaDisableFeatureActionRequest;
declare type IMcaReadyForOperationActionRequest =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaReadyForOperationActionRequest;
declare type IMcaReadyForOperationActionResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaReadyForOperationActionResponse;
declare type IMcaReadyForOperationActionResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaReadyForOperationActionResponseData;
declare type IMcaOnDataUpdatedEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnDataUpdatedEventResponse;
declare type IMcaOnDataUpdatedData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnDataUpdatedData;
declare type IMcaOnDataUpdatedOutData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnDataUpdatedOutData;
declare type IMcaEventData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaEventData;
declare type IMcaOnDataUpdatedEventData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnDataUpdatedEventData;
declare type IMcaOnDataUpdatedOutDataResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnDataUpdatedOutDataResponse;
declare type IMcaonOutgoingEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaonOutgoingEventResponse;
declare type IMcaOnOutgoingEventData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnOutgoingEventData;
declare type IMcaOnOutgoingEventPayloadData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnOutgoingEventPayloadData;
declare type IMcaOnOutgoingEventOutData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnOutgoingEventOutData;
declare type IMcaOnOutgoingEventOutDataResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnOutgoingEventOutDataResponse;
declare type IMcaAgentStateEventActionResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaAgentStateEventActionResponseData;
declare type IMCAAgentStateEventResponsePayload =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMCAAgentStateEventResponsePayload;
declare type IMcaAgentStateEventActionResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaAgentStateEventActionResponse;
declare type IMcaAgentStateEventActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaAgentStateEventActionRequest;
declare type IMcaOutBoundCommErrorActionResponseData =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOutBoundCommErrorActionResponseData;
declare type IMcaOutBoundCommErrorActionResponse =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOutBoundCommErrorActionResponse;
declare type IMCAOutBoundCommErrorResponsePayload =
    CX_SVC_UI_EVENTS_FRAMEWORK.IMCAOutBoundCommErrorResponsePayload;

```



```
declare type IMcaOutboundCommErrorActionRequest =
  CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOutboundCommErrorActionRequest;
declare type IMcaEventRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaEventRequest;
declare type IMcaOnToolbarInteractionCommandEventResponse =
  CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnToolbarInteractionCommandEventResponse;
declare type IMcaOnToolbarInteractionCommandDataResponse =
  CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnToolbarInteractionCommandDataResponse;
declare type IMcaOnToolbarInteractionCommandData =
  CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnToolbarInteractionCommandData;
declare type IMcaOnToolbarAgentCommandEventResponse =
  CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnToolbarAgentCommandEventResponse;
declare type IMcaOnToolbarAgentCommandDataResponse =
  CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnToolbarAgentCommandDataResponse;
declare type IMcaOnToolbarAgentCommandData = CX_SVC_UI_EVENTS_FRAMEWORK.IMcaOnToolbarAgentCommandData;
declare type ISetChannelAvailabilityResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ISetChannelAvailabilityResponse;
declare type ISetChannelAvailabilityRequest = CX_SVC_UI_EVENTS_FRAMEWORK.ISetChannelAvailabilityRequest;
declare type ISetToolbarPropertiesActionRequest =
  CX_SVC_UI_EVENTS_FRAMEWORK.ISetToolbarPropertiesActionRequest;
declare type ISetToolbarPropertiesActionResponseData =
  CX_SVC_UI_EVENTS_FRAMEWORK.ISetToolbarPropertiesActionResponseData;
declare type ISetToolbarPropertiesActionResponse =
  CX_SVC_UI_EVENTS_FRAMEWORK.ISetToolbarPropertiesActionResponse;
declare type IInteractionLogger = CX_SVC_UI_EVENTS_FRAMEWORK.IInteractionLogger;
declare type IInteractionRecorder = CX_SVC_UI_EVENTS_FRAMEWORK.IInteractionRecorder;
declare type IExecuteSmartAction = CX_SVC_UI_EVENTS_FRAMEWORK.IExecuteSmartAction;
declare type ISmartActionOperationResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ISmartActionOperationResponse;
declare type ISmartActionOperationResponseData =
  CX_SVC_UI_EVENTS_FRAMEWORK.ISmartActionOperationResponseData;
declare type ISmartActionContext = CX_SVC_UI_EVENTS_FRAMEWORK.ISmartActionContext;
declare type IRecordState = CX_SVC_UI_EVENTS_FRAMEWORK.IRecordState;
declare type ISmartActionEventRequest = CX_SVC_UI_EVENTS_FRAMEWORK.ISmartActionEventRequest;
declare type ISmartActionEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.ISmartActionEventResponse;
declare type ISmartActionEventData = CX_SVC_UI_EVENTS_FRAMEWORK.ISmartActionEventData;
declare type IComposeMessageEventRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IComposeMessageEventRequest;
declare type IComposeMessageEventResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IComposeMessageEventResponse;
declare type IComposeMessageEventData = CX_SVC_UI_EVENTS_FRAMEWORK.IComposeMessageEventData;
declare type IComposeMessageActionRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IComposeMessageActionRequest;
declare type IComposeMessageDataRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IComposeMessageDataRequest;
declare type IComposeMessageDataResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IComposeMessageDataResponse;
declare type IComposeMessageData = CX_SVC_UI_EVENTS_FRAMEWORK.IComposeMessageData;
declare type ISpecialFieldMapEntry = CX_SVC_UI_EVENTS_FRAMEWORK.ISpecialFieldMapEntry;
declare type IGetGenAiRequest = CX_SVC_UI_EVENTS_FRAMEWORK.IGetGenAiRequest;
declare type IGetGenAiActionResponse = CX_SVC_UI_EVENTS_FRAMEWORK.IGetGenAiActionResponse;
declare type IGetGenAiActionData = CX_SVC_UI_EVENTS_FRAMEWORK.IGetGenAiActionData;
declare type IOnWrapUpTimerExpiredEventResponse =
  CX_SVC_UI_EVENTS_FRAMEWORK.IOnWrapUpTimerExpiredEventResponse;
declare type IEngagementData = CX_SVC_UI_EVENTS_FRAMEWORK.IEngagementData;
```

Because the compiler is started in watch mode (`tsc -w`), the `main.js` file will be autogenerated on saving `main.ts`. Reload the client application by right-clicking it and selecting **Reload Frame** from the drop-down list.

To test go to the Fusion application home page and a service request. Keep the developer console open. Now switch the MSI tabs. The `tabChangeResponse`, `currentTabContext`, and `previousTabContext` details will be logged into the console on the tab switch.

Publish the Set Field Value Operation

Update the request type and subscription in the above code as given in the following example. Then, transpile the code and reload the iframe.

```
async function run() {
  const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('appName', 'v1');
  const tabContext: ITabContext = await frameworkProvider.getTabContext();
```

```
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const requestObject: ISetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation') as
ISetFieldValueOperationRequest);
requestObject.field().setValue('ServiceRequest.Title', 'New Title');
// publish event is happening here.
// The result of the async operation is available on then(for success) and on catch ( for failure)
recordContext.publish(requestObject).then((message) => {
const response = message as ISetFieldValueResponse;
console.log(response);
}).catch((error: IErrorData) => {
console.log(error);
});
}
```

Load the UI Events Framework Client Library

You can load the UEF client library into your third-party application by adding the following tag into the head section:

```
<script src = "https://static.oracle.com/cdn/ui-events-framework/libs/ui-events-framework-client.js"></script>
```

With this, CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework will be available in the application's document object model.

Initialize the Framework

You can use the initialize API provided by CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework to initialise UEF in your third-party application:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
```

Once initialized, you can use the frameworkProvider frame:

```
interface IUiEventsFrameworkProvider {
requestHelper: IRequestHelper;
dispose: () => void;
getVersion(): string;
getApplicationName(): string;
getCurrentTabId(): Promise<string>;
getActiveTabId(): Promise<string>;
getAvailableRecords(tabId?: string): Promise<IRecordContext[]>;
getActiveTab(): Promise<ITabContext>;
getTabContext(browserTabId?: string): Promise<ITabContext>;
getAvailableTabs(): Promise<ITabContext[]>;
getGlobalContext(): Promise<IGlobalContext>;
getModalWindowContext(): Promise<IModalWindowContext>;
getSidePaneContext(sidePaneId: string): Promise<ISidePaneContext>;
}
```

Check [Integrate an External Application with Service Center](#) for instructions how to use it and reference for a detailed explanation of each method available in the interface.

Note: The Dispose method is available in frameworkProvider. When called, all the currently active subscriptions are destroyed. You application must initialize (CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize) again to use the UEF.

Get the Corresponding Context

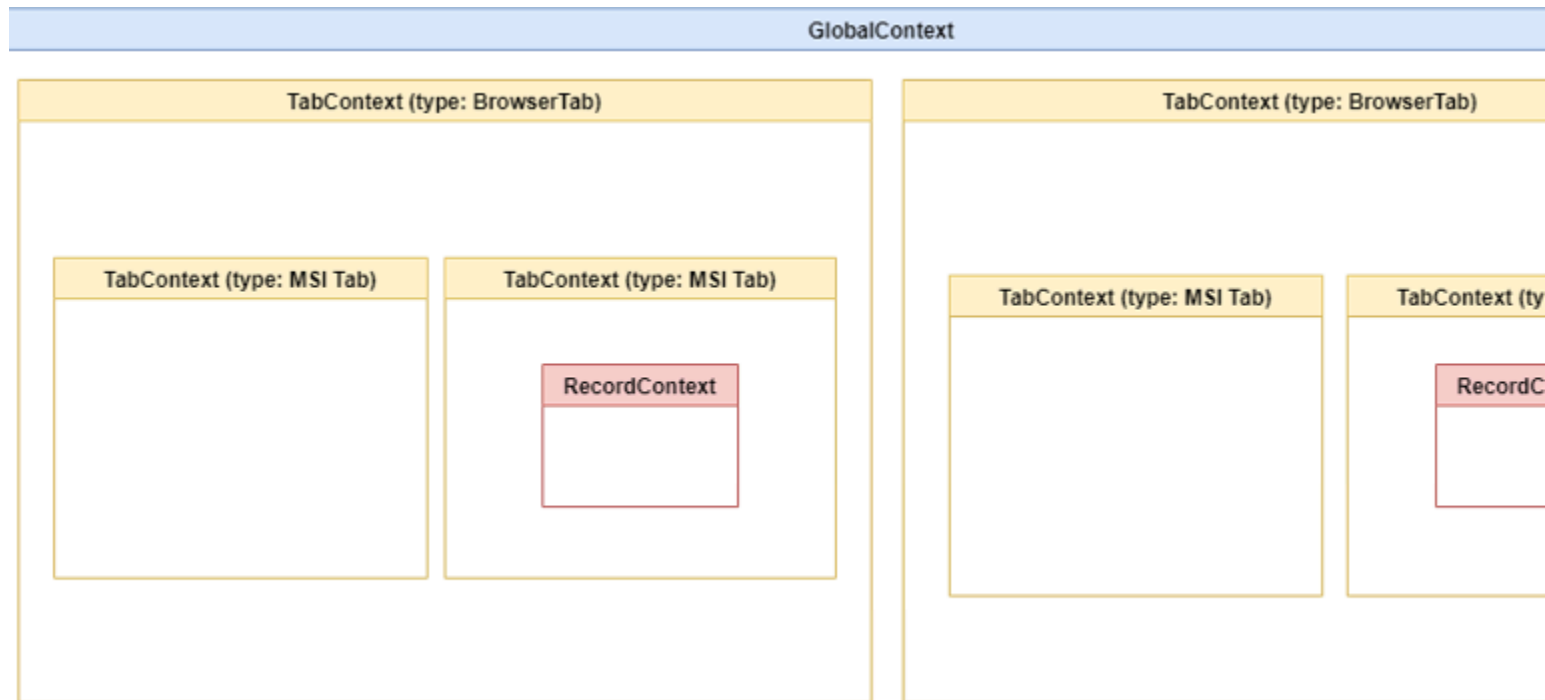
Extensible events and operations of UEF are supported on top of contexts like GlobalContext, TabContext, and RecordContext.

The following diagram details the hierarchy of these contexts that are maintained in UEF and the supported events, actions, and APIs supported in each context.

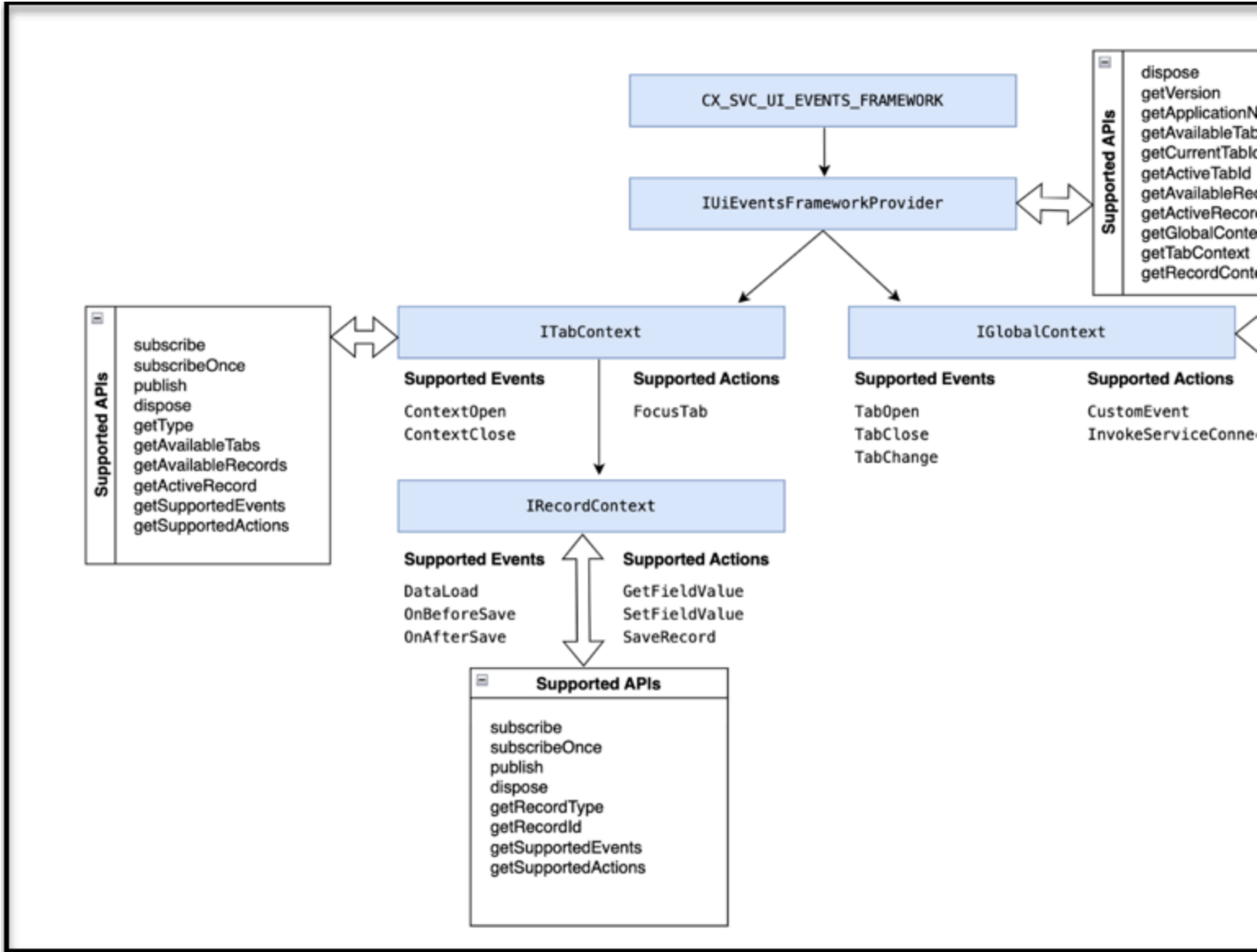
First, here's how you get the various contexts:

```
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();  
const tabContext: ITabContext = await frameworkProvider.getTabContext(browserTabId);  
const availableRecords: IRecordContext[] = await frameworkProvider.getAvailableRecords();
```

The following graphic illustrates the global context:



And this graphic shows the overall architecture:



<INSERT 2 GRAPHICS--UEF_Global_context AND UEF_diagram

Here's a table explaining the context types:

Context Type	Support Events	Support Operations	Syntax
GlobalContext	TabOpen, TabClose, TabChange	PopOperation, InvokeServiceConnection	getGlobalContext(): Promise<IGlobalContext>;
TabContext	ContextOpen, ContextClose	FocusTab, CloseTab	getTabContext(browserTabId?: string): Promise<ITabContext>;

Context Type	Support Events	Support Operations	Syntax
RecordContext	FieldValueChange, OnBeforeSave(Controllable Event), OnAfterSave, OnDataLoad	GetFieldValue, SaveRecord, SetFieldValue	getAvailableRecords(tabId?: string): Promise<IRecordContext[]>;

Note: The OnBeforeSave (in Record Context) event is fired before firing the REST API that commits the data into the server. End-users can do asynchronous operations in this callback, or they can cancel this event. This event can wait for an asynchronous operation to be completed.

Generate the Request Object

The requestObject is used by the publish or subscribe API.

The request object has information about the name of the interested event or operation and the data required for event subscription or publish operation. For example, say you want to update the title field. For that, you create a request object as shown in the following example and publish it with the publish API: `publish ( requestObject )`.

```
requestObject = frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation')
requestObject.field().setField('ServiceRequest.Title').setValue('SR1') // setting a value
```

Now, say you wanted to get notified of changes in the title and problem description field. In this case, you must create the request object with details of the event name, field name title, and problem description as shown in the following example:

```
requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusFieldValueChangeEvent')
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription'])
// interested fields to be subscribed
```

The usage of the requestObject's exposed methods varies with events and operations. The following sample shows some scenarios of requestObject:

```
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const tabContext: ITabContext = await frameworkProvider.getTabContext(browserTabId);
const availableRecords: IRecordContext[] = await frameworkProvider.getAvailableRecords();
```

Invoke the Publish or Subscribe API on the Context

Publish is used to initiate an operation as shown in the following example:

```
<context>.publish (requestObject)
```

Subscribe is used to register a callback for an event from Service Center. It's like an event listener. When the event is generated in the Service Center, a corresponding callback, if registered, will be called in the UEF. SubscribeOnce is similar to subscribe. The difference is once the event is generated in the Service Center, the registered call back will be called by UEF, and it then disposes of the subscription. Here's an example:

Subscribe to an Event	Subscribe Once to an Event
<pre><context>.subscribe(requestObject, function callback(response) { console.log(response) })</pre>	<pre><context>.subscribeOnce(requestObject, function callback(response) { console.log(response) })</pre>

Subscribe and SubscribeOnce returns a subscription object that must be deleted if the subscription is no longer in need, or else it will stay in the UEF as a garbage yet active subscription

Integrate an External Application with Service Center

UEF enables users to consume extensible events and operations inside their applications through dedicated APIs.

The following topics will help you integrate your external application with the Service Center so you can listen to Service Center-originated events and perform specific actions to the Service Center application.

Load the External Application Inside Service Center

You can load an external application in the Service Center using the custom component `oj-cx-svc-common-ui-events-container`.

Find this component on Oracle Component Exchange.

Once you have installed this custom component VB Studio instance, you can add it to any dynamic container that's available in the Service Center application. Here are the steps:

1. Install the custom component `oj-cx-svc-common-ui-events-container` from Oracle Component Exchange.
2. Add the component to the desired area in Service Center.
3. Provide your external application URL in the URL property of the `oj-cx-svc-common-ui-events-container` component.

Load UI Events Framework Inside an External Application

For an external application to interact with the Service Center, the external application must be added to the UI Events Framework library file in the external application itself.

The library file is available at <https://static.oracle.com/cdn/ui-events-framework/client/ui-events-framework-client.js>. Add the UI Events Framework library file to the external application using any native method of loading scripts into an HTML file, such as loading the library file into an HTML file using a script tag as the following example show:

```
<script src="https://static.oracle.com/cdn/ui-events-framework/libs/ui-events-framework-client.js"></script>
```

Type declarations are available in `uiEventsFramework`: <https://static.oracle.com/cdn/ui-events-framework/libs/23.01.1.5/uiEventsFramework.d.ts>. Add this file to the TypeScript project for type checking.

Once the library is loaded in the external application following object will be available in the application's DOM.

```
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework
```

Users can use APIs exposed the object to interact with the Service Center.

How to Use UI Events Framework

To expose an API in the UI Events Framework, here are the general steps to follow:

Prerequisites

1. Load the external application in the Service Center using the `oj-cx-svc-common-ui-events-container` component.
2. Load UI Events Framework library file in external application's HTML or DOM file.

Overview of the Steps

1. Initialize the framework.
2. Get the required context for the use case.
3. Generate request object using `requestHelper`.
4. Invoke the APIs exposed in the above contexts.

Initialize UI Events Framework

Now you initialize the framework using an application name for any interaction. Once the framework is initialized successfully, you can get the reference to the instance of `IUiEventsFrameworkProvider`. All the interactions happening from this object can be tracked with the application name provided during the initialization.

Here's an example of the syntax for initializing the UI Events Framework:

```
initialize(applicationId: string, version?: string): Promise<IUiEventsFrameworkProvider>;
```

Parameters

Parameter Name	Required	Description
applicationId	Yes	Users must call the initialize function with a unique application ID to use the UI Events Framework. The application ID helps debug purposes when testing.
version	No	Oracle aims maintain backward compatibility for all the releases. If in the future, the framework deprecates some APIs without backward compatibility, use this parameter to load older or newer versions of the library. Currently, the version value defaults to v1.

The following code example shows an example in TypeScript:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: CX_SVC_UI_EVENTS_FRAMEWORK.IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');
```

The following code example shows an example in JavaScript:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');
```

Retrieve the Corresponding Context for the Use Case

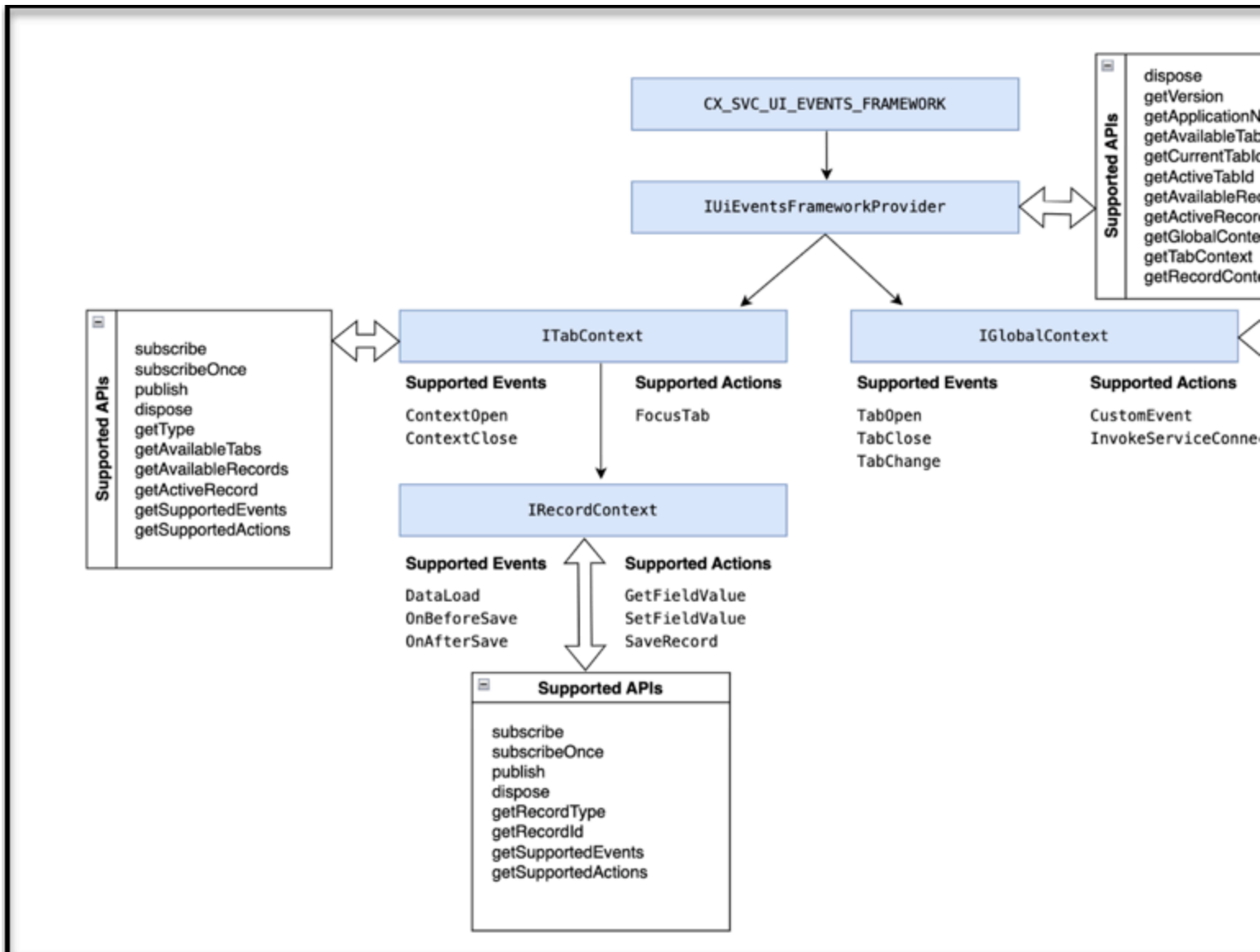
Once you can retrieve the instance of `IUiEventsFrameworkProvider`, you can now choose the context based on use cases.

The `UiEventsFrameworkProvider` provides three contexts as shown in the following:

- `GlobalContext`
- `TabContext`
- `RecordContext`

All the extensible events and actions of UEF are supported on top of contexts like `GlobalContext`, `TabContext`, and `RecordContext`. The following diagram shows the hierarchy of these contexts that are maintained in UEF and the supported events, actions, and APIs supported in each context:

The following graphic shows the overall architecture:



Initialize the Framework

You can use the initialize API provided by CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework to initialise UEF in your third-party application:

```

/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');

```

Once initialized, you can use the frameworkProvider frame:

```

interface IUiEventsFrameworkProvider {
  requestHelper: IRequestHelper;
  dispose: () => void;
  getVersion(): string;
}

```

```

getApplicationName(): string;
getCurrentTabId(): Promise<string>;
getActiveTabId(): Promise<string>;
getAvailableRecords(tabId?: string): Promise<IRecordContext[]>;
getActiveTab(): Promise<ITabContext>;
getTabContext(browserTabId?: string): Promise<ITabContext>;
getAvailableTabs(): Promise<ITabContext[]>;
getGlobalContext(): Promise<IGlobalContext>;
getModalWindowContext(): Promise<IModalWindowContext>;
getSidePaneContext(sidePaneId: string): Promise<ISidePaneContext>;
}

```

Check [Integrate an External Application with Service Center](#) for instructions how to use it and reference for a detailed explanation of each method available in the interface.

Note: The Dispose method is available in frameworkProvider. When called, all the currently active subscriptions are destroyed. You application must initialize (CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize) again to use the UEF.

GlobalContext

GlobalContext enables you to listen to application-level events and perform application-level operations.

For example, it enables listening to tabOpen, tabClose and tabChange events. It also supports TabFocus, TabClose, CustomEvent, and InvokeRestApi Actions.

Supported events are as follows:

Event Name	Description
TabOpen	This event is initiated when a new browser tab or a MSI tab is opened.
TabClose	This event is initiated when a browser tab or MSI tab is closed
TabChange	This event is initiated when a browser tab or MSI tab switch happens

To get the application reference to GlobalContext, you can use the getGlobalContext function.

The following code sample provides a reference to the application:

```
getGlobalContext(): Promise<IGlobalContext>;
```

The following code sample shows an example in TypeScript:

```

/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider : CX_SVC_UI_EVENTS_FRAMEWORK.IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');
const globalContext: CX_SVC_UI_EVENTS_FRAMEWORK.IGlobalContext = await frameworkProvider.getGlobalContext();

```

The following code sample shows an example in JavaScript:

```

const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');

```



```
const globalContext = await frameworkProvider.getGlobalContext();
```

TabContext

TabContext provides the ability to listen to tab-level events such as ContextOpen, ContextClose and perform tab-level operations such as FocusTab and CloseTab.

Supported events are as follows:

Event Name	Description
ContextOpen	This event is initiated when a new object gets opened in a particular tab.
ContextClose	This event is initiated when an object gets closed in a particular tab.

Here are the supported actions:

Action Name	Description
FocusTab	This action is used to focus on a particular browser or MSI tab.
CloseTab	This action is used to close a particular browser or MSI tab.

To get the reference of TabContext, you can use the getTabContext function. This function provides a reference to the active tab in which the application is loaded. In addition, it can pass the browserTabId as an optional parameter to get a particular browser tab's context.

Here's the syntax:

```
getTabContext(browserTabId?: string): Promise<ITabContext>;
```

Here's the parameters:

Parameter Name	Required?	Description
browserTabId	No	A particular browser tab's id. Note: You can get each browser's tabId by calling the getCurrentTabId API on frameworkProvider from each browser tab.

The following code sample shows an example in TypeScript:

```
/// <reference path="uiEventsFramework.d.ts"/>  
const frameworkProvider : IUiEventsFrameworkProvider = await
```

```
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');  
const tabContext: ITabContext = await frameworkProvider.getTabContext();
```

The following code sample shows an example in JavaScript:

```
const frameworkProvider = await  
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');  
const tabContext = await frameworkProvider.getTabContext();
```

RecordContext

RecordContext enables you to listen to all events or operations related to an object. For example, listening to field value changes before an event is saved or after an event is saved, and so on.

Here are the supported events:

Event Name	Description
Field Value Change	This event is initiated when a field value is changed. For example, an SR title is changed.
On Before Save	This event is initiated before firing the API request to commit the data to the server. The end-user should be able to do asynchronous operations in this callback to cancel the event. This event can wait for an asynchronous operation to be completed
On After Save	This event is initiated after a record is successfully saved in the VB application. For example, an SR is saved.
On Data Load	This event is initiated when data is loaded. For example, after the SR object data is fetched.

Here are the supported actions:

Action Name	Description
GetFieldValue	Used to fetch the current value of a field. For example, retrieving the current value of the Title field of SR.
Save Record	Used to save an open record. For example, save an SR form from the edit page or save an SR from the create form.
UpdateFieldValue	Used to update the value of a particular field. For example, update the Title field of an SR.

You get the reference of RecordContext only by calling `getActiveRecord()` API on top of TabContext object, which can be either a browser-TabContext or a MSI-TabContext. Apart from `getActiveRecord` API, you can get the RecordContext object as a response of ContextOpen event subscription.

Generate a Request Object Using requestHelper

To subscribe to app-level events or to perform app-level operations, the external application must be able to invoke subscribe and publish APIs provided by the UI Events Framework.

These APIs will expect a request object which will carry the information about the events to be listened to or operations to be performed. The framework provides a requestHelper object which will enable the user to generate the request object specific to each event or operation.

The following code example shows the syntax for requestHelper object:

```
IRequestHelper {
  createSubscriptionRequest(eventName: string): IEventRequest;
  createPublishRequest(operationName: string): IOperationRequest;
}
IEventRequest {
  getEventName(): string;
}
IOperationRequest {
  getOperationName(): string;
}
```

The following TypeScript example shows how you generate a request object for an event subscription:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider =
  await CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IEventRequest =
  frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabOpenEvent');
globalContext.subscribe(requestObject, (response: IEventResponse) => {
  // custom code
});
```

The following JavaScript code sample shows how to generate request object for event subscription:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabOpenEvent');
globalContext.subscribe(requestObject, (response) => {
  // custom code
});
```

The following TypeScript code sample shows how to generate request object for performing an operation:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const restCallRequest: IServiceConnectionRequest =
  (frameworkProvider.requestHelper.createPublishRequest('InvokeServiceConnection') as
  IServiceConnectionRequest);
restCallRequest.setServiceConnectionId('interactions/update_interactions');
restCallRequest.setParameters({ "interactions_Id": "12345" });
restCallRequest.setBody({ "StatusCd": "ORA_SVC_CLOSED" });
globalContext.publish(restCallRequest).then((message: IOperationResponse) => {
  // custom code
}).catch((error: IErrorData) => {
  // custom code
});
```

```
});
```

The following JavaScript code sample shows how to generate request object for performing an operation:

```
const frameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID');
const globalContext = await frameworkProvider.getGlobalContext();
const restCallRequest = (frameworkProvider.requestHelper.createPublishRequest('InvokeServiceConnection'));
restCallRequest.setServiceConnectionId('interactions/update_interactions');
restCallRequest.setParameters({ "interactions_Id": "12345" });
restCallRequest.setBody({ "StatusCd": "ORA_SVC_CLOSED" });
globalContext.publish(restCallRequest).then((message) => {
  // custom code
  const response = message.getResponseData();
  console.log(response.getStatus());
  console.log(response.getBody());
}).catch((error) => {
  // custom code
});
```

Invoke the APIs

The APIs previously discussed are Subscribe, SubscribeOnce, Publish, and Dispose.

Here's an overview of each:

Subscribe

This API can subscribe to an event from Service Center. Using this API, the external application can listen to the application or object level events from the Service Center. Once an event is subscribed to using this API, an external application is notified until the subscription is disposed.

The following code sample shows the syntax for Subscribe:

```
subscribe: (payload: IEventSubscriptionPayload, callbackFunction: (response: IEventResponsePayload) => void)
=> ISubscriptionContext;
```

Parameter	Required?	Description
payload	Yes	Requests details for the subscription with with corresponding event name.

SubscribeOnce

Use this API to subscribe to a Service Center event just once. Using this API, external applications can listen to the application or object level events from Service Center. This subscription is disposed automatically after the first notification.

The following code sample shows the syntax for SubscribeOnce:

```
subscribeOnce: (payload: IEventSubscriptionPayload, callbackFunction: (response: IEventResponsePayload) =>
void)
=> ISubscriptionContext;
```

Parameter	Required?	Description
payload	Yes	Requests details for the subscription with with corresponding event name.
callbackFunction	Yes	A callback function, which receives calls whenever events are triggered within Service Center.

Publish

This API can inform the Service Center to operate. The API returns a promise to which you can add a then-catch block. The block then returns the operation's status and any data returned after the process is completed.

The following code sample shows the syntax for Publish:

```
publish: (payload: IOperationPublishPayload) => Promise<IOperationResponsePayload>;
```

Parameter	Required?	Description
payload	Yes	Requests details for the subscription with with corresponding operationName.

Dispose

This API can dispose or unregister any subscriptions that's added for an event. By disposing the subscription, the API will not trigger any further notifications to the client side receiver.

The following code sample shows the syntax for Dispose:

```
dispose: () => void;
```

Supported Events

The events emitted by the UI Events Framework are classified into the following divisions:

- Controllable Events
- Non-Controllable Events

Controllable Events

You have the option of canceling or deferring these events for a certain period of time.

Event	Context	Description
OnBeforeSave	RecordContext	This event is triggered before the API request to commit the data to the server is initiated. You can do asynchronous operations in this callback or Cancel event. This event can wait for the completion of an asynchronous operation.

Non-Controllable Events

These events are different from controllable events, and can't be controlled in the same way. These events are similar to notification events in that they notify the client when an event occurs in the application.

Events	Context	Description
TabOpen	GlobalContext	This event is initiated when a browser tab or the MSI tab gets opened with the Service Center application.
TabClose	GlobalContext	This event is initiated when a browser tab or the MSI tab gets closed with the Service Center application.
TabChange	GlobalContext	This event is initiated when a browser tab or the MSI tab happens with the Service Center application.
ContextOpen	GlobalContext	This event is initiated when an object is opened in the UI. For example, an SR is opened.
DataLoad	RecordContext	This event is initiated when data is loaded. For example, after fetching SR object data.
ContextClose	GlobalContext	This event is initiated when an object is closed in the UI. For example, an SR is closed.
FieldValueChange	RecordContext	This event is initiated when a field value change occurs. For example, an SR title is changed.
OnAfterSave	RecordContext	This event is initiated after saving an open object in the UI. For example, after saving an SR.

Operations

You can use extensibility operations to execute specific tasks in the Service Center application. The operations that are supported are listed in the following table:

Operation	Syntax	Description
PopOperation	GlobalContext	To pop a new browser tab, MSI tab, or any custom page in any desired tab.
FocusTab	TabContext	To focus on a particular MSI tab or browser tab.
CloseTab	TabContext	To close a particular MSI tab or browser tab.
InvokeServiceConnection	GlobalContext	To invoke a service defined in the application and fetch its response.
SetFieldValue	RecordContext	To update the value of a particular field. For example, update the Title field of SR.
GetFieldValue	RecordContext	To fetch the current value of a field. For example, to get the current value of the Title field of SR.
SaveRecord	RecordContext	To save an open record. For example: Save an SR form from edit page or save an SR from create form.

Supported Operations

You can use extensibility operations to execute certain tasks within the application. The following table lists those operations you can use:

Operation	Context	Description
PopOperation	GlobalContext	To pop a new browser tab, MSI tab, or any custom page in any desired tab.
FocusTab	TabContext	To focus on a particular MSI tab or browser tab.
CloseTab	TabContext	To close a particular MSI tab or browser tab.
InvokeServiceConnection	GlobalContext	To invoke a service defined in the application and fetch its response.
SetFieldValue	RecordContext	Use to update the value of a particular field. For example, to update the Title field of an SR.
GetFieldValue	RecordContext	Use to fetch the current value of a field. For example, to get the current value of the Title field of an SR.

Operation	Context	Description
Save Record	RecordContext	Use to save an open record. For example, save an SR form from the Edit page or save an SR from Create form.

Events Subscription

TabOpen Event

TabOpen event subscription gives a notification about a browser tab or new window tab open event, or a new MSI tab open event happening in any already opened browser tabs with the Fusion application.

The event response will give the TabContext of the opened tab, on top of which you can call APIs to subscribe or publish APIs or any other tabContext supported APIs. Also, by calling getType API on tabContext, you can identify the type of the opened tab. The type could be the Browser tab, MSI tab, or MSI sub tab.

Note: TabOpen is an event listenable from GlobalContext.

Here's a TypeScript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');

const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();

const payload: IEventRequest =
  frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabOpenEvent');
globalContext.subscribe(payload, (response: IEventResponse) => {
  const tabOpenResponse = response as ITabEventResponse;
  const tabContext: ITabContext = tabOpenResponse.getResponseData();
  const type = tabOpenResponse.getType();
})
```

Here's a JavaScript example:

```
const frameworkProvider =await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const payload =
  frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabOpenEvent');
globalContext.subscribe(payload, (response) => {
  const tabContext = response.getResponseData();
  const type = tabOpenResponse.getType();
})
```

TabClose Event

The TabClose event subscription gives a notification about a browser tab, a new window tab close event, or a new MSI tab close event happening in any already opened browser tabs with the Fusion application loaded.

The event response will give the ITabInfo object of the closed tab, on top of which you can call gettabld(), getMSITabld(), and getMSISubTabld() to get the browser tabld, MSI tabid, and MSI sub tab id of the closed tab respectively.

Note: TabClose is an event listenable from GlobalContext.

Here's a TypeScript example:

```
const subscribeTabClose = async () => {

    const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('appname', 'v1');
    const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();

    const payload: IEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabCloseEvent');
    globalContext.subscribe(payload, (response: IEventResponse) => {
        let responseData = response as ITabCloseEventResponse;
        console.log(responseData.getResponseData().getTabId()); // Opened Tab's identifier
        console.log(responseData.getEventName()); // 'cxEventBusTabCloseEvent'
        console.log(responseData.getContext()); // contextObject info
    });
}
```

Here's a JavaScript example:

```
const subscribeTabClose = async () => {

    const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('appname', 'v1');
    const globalContext = await frameworkProvider.getGlobalContext();
    const payload = frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabCloseEvent');
    globalContext.subscribe(payload, (responseData) => {
        console.log(responseData.getResponseData().getTabId()); // Opened Tab's identifier
        console.log(responseData.getEventName()); // 'cxEventBusTabCloseEvent'
        console.log(responseData.getContext()); // contextObject info
    });
}
```

TabChange Event

The TabChange event subscription gives a notification about a browser tab or an MSI tab switch happening in the Fusion application.

The user receives a notification of all tab-to-focus-out events once it adds a subscription to this event. It gives the newly focused tab's context in `getCurrentTab()` and the previous tab's context in `getPreviousTab` methods in the event response.

Note: TabChange is an event listenable from GlobalContext.

Here's a TypeScript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('appname', 'v1');
    const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
    const payload: IEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabChangeEvent');
    globalContext.subscribe(payload, (response: IEventResponse) => {
        let responseData = response as ITabChangeEventResponse;
        let tabChangeResponse: ITabChangeEventResponse = responseData.getResponseData();
        let currentTabContext: ITabContext = tabChangeResponse.getCurrentTab();
        let previousTabContext: ITabContext = tabChangeResponse.getPreviousTab();
    });
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('appname', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const payload =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusTabChangeEvent');
globalContext.subscribe(payload, (responseData) => {
  let tabChangeResponse = responseData.getResponseData();
  let currentTabContext = tabChangeResponse.getCurrentTab();
  let previousTabContext = tabChangeResponse.getPreviousTab();
});
```

ContextOpen Event

This event is started when an object is opened in a VB application. In the event response, you can get the RecordContext object of the opened object and call all the object-related actions and events, such as setFieldValue, getFieldValue, fieldValueChange, and so on top of the RecordContext object.

Note: ContextOpen is an event listenable from TabContext.

The following code sample shows an example in TypeScript for subscribing to ContextOpen event:

```
/// <reference path="uiEventsFramework.d.ts"/>

const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getCurrentBrowserTabContext();
const requestObject: IEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusContextOpenEvent');
tabContext.subscribe(requestObject, (message: IEventResponse) => {
  const response = message as IContextOpenEventResponse;
  const recordContext: IRecordContext = response.getResponseData();
});
```

The following code sample shows an example in JavaScript for subscribing to ContextOpen event:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext = await frameworkProvider.getCurrentBrowserTabContext();
const requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusContextOpenEvent');
tabContext.subscribe(requestObject, (message) => {
  const response = message;
  const recordContext = response.getResponseData();
});
```

Context Close Event

This event is started when an object is closed in a VB application. This is a global-level event, so the user must call the subscribe API on top of global context to subscribe to this event.

Note: ContextClose is an event listenable from TabContext.

The following code sample shows an example in TypeScript for subscribing to ContextClose event:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
```

```
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const requestObject: IEventRequest =
  frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusContextCloseEvent');
tabContext.subscribe(requestObject, (message: IEventResponse) => {
  const response = message as IContextResponse
  const context: IObjectContext = response.getResponseData();
  console.log(context.getObjectType());
  console.log(context.getObjectId());
});
```

The following code sample shows an example in JavaScript for subscribing to ContextClose event:

```
const frameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext = await frameworkProvider.getCurrentBrowserTabContext();
const requestObject =
  frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusContextCloseEvent');
tabContext.subscribe(requestObject, (message) => {
  const response = message;
  const context = response.getResponseData();
  console.log(context.getObjectType());
  console.log(context.getObjectId());
});
```

DataLoad Event

This event is started when data is loaded for a particular object in a VB application. This event will always be fired after the ContextOpen event notification of a particular object. This is a record-specific event.

Note: DataLoad is an event listenable from RecordContext.

The following code sample shows an example in TypeScript for subscribing to DataLoad event:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const requestObject: IEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusDataLoadEvent');
recordContext.subscribe(requestObject, (response: IEventResponse) => {
  const message = response as IContextResponse;
  // custom code
});
```

The following code sample shows an example in JavaScript for subscribing to DataLoad event:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();
const requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusDataLoadEvent');
recordContext.subscribe(requestObject, (response) => {
  // custom code
});
```

Field Value Change Event

Here are examples of subscribing to FieldValueChange event where field names are passed.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();

const requestObject: IFieldValueChangeEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusFieldValueChangeEvent') as
IFieldValueChangeEventRequest;
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.subscribe(requestObject, (message: IEventResponse) => {
const response = message as IFieldValueChangeEventResponse;
const fieldName: string = response.getResponseData().getFieldName();
const newValue: string | boolean | number = response.getResponseData().getNewValue();
const oldValue: string | boolean | number = response.getResponseData().getOldValue();
});
```

The following code sample shows an example in JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();
const requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusFieldValueChangeEvent')
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.subscribe(requestObject, (response) => {
const fieldName = response.getResponseData().getFieldName();
const newValue = response.getResponseData().getNewValue();
const oldValue = response.getResponseData().getOldValue();
});
```

On After Save Event

This event is started after successfully saving a record in the VB application. This is a record level event, so the user must call the subscribe API on top of record context to subscribe to it.

This event response gives information about the saved object's type and its old and new identifiers. The old identifier and new identifier is different only when a save event happens on a new object. For example, while creating a new object UEF assigns a `-ve` identifier to it, and this `-ve` identifier is received in the ContextOpen event's subscription response. After saving, this object is assigned an actual identifier, and the new or actual identifier value can be tracked by adding the OnAfterSave Event listener to this new object.

Note: OnAfterSave is an event listenable from RecordContext.

The following code sample shows an example in TypeScript for subscribing to OnAfterSave event:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();
const requestObject: IEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusOnAfterSaveEvent');
recordContext.subscribe(requestObject, (message: IEventResponse) => {
const response: IOnAfterSaveEventResponse = message as IOnAfterSaveEventResponse;
const oldObjectId: string = response.getResponseData().getOldObjectId();
const newObjectId: string = response.getResponseData().getObjectId();
const objectType: string = response.getResponseData().getObjectType();
});
```

```
});
```

The following code sample shows an example in JavaScript for subscribing to OnAfterSave event:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();

const requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusOnAfterSaveEvent');
recordContext.subscribe(requestObject, (response) => {
const oldObjectId = response.getResponseData().getOldObjectId();
const newObjectId = response.getResponseData().getObjectId();
const objectType = response.getResponseData().getObjectType();
});
```

OnBeforeSave Event

This event is started before the API request to commit the data to the server. You can then do asynchronous operations in this callback or cancel this event. This event can wait for an asynchronous operation to be completed. With this event, the user will get complete control of the save event that happens in VB by either approving or canceling this operation in VB.

Note: OnBeforeSave is an event listenable from RecordContext

The following code sample shows an example in TypeScript for subscribing to OnBeforeSave event:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();

const requestObject: IEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusOnBeforeSaveEvent');

recordContext.subscribe(requestObject, (response: IEventResponse) => {
// custom code
return new Promise((resolve, reject) => {
});
// the VB application save can be controlled by either resolving or rejecting this promise.
// Resolving it would allow the save process, and rejecting the promise will cancel
the save process inside VB.
// The save process in VB will wait until this promise is resolved or rejected.
});
```

The following code sample shows an example in JavaScript for subscribing to OnBeforeSave event:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();
const requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusOnBeforeSaveEvent');
recordContext.subscribe(requestObject, (response) => {
// custom code
return new Promise((resolve, reject) => {
});
// the VB application save can be controlled by either resolving or rejecting this promise.
// Resolving it would allow the save process and rejecting the promise will cancel the save process inside
VB.
// The save process will wait until this promise is resolved or rejected.
```

```
});
```

Operations

InvokeServiceConnection Operation

Use this operation to call a service connection or REST call through the UI Events Framework.

Note: InvokeServiceConnection Operation is an operation publishable from the GlobalContext level. Currently, only Internal Service connections marked as extensible are supported.

Here's a TypeScript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();

const restCallRequest: IServiceConnectionRequest =
  (frameworkProvider.requestHelper.createPublishRequest('InvokeServiceConnection') as
  IServiceConnectionRequest);
restCallRequest.setServiceConnectionId('interactions/update_interactions');
restCallRequest.setParameters({ "interactions_Id": "12345" });
restCallRequest.setBody({ "StatusCd": "ORA_SVC_CLOSED" });

globalContext.publish(restCallRequest).then((message: IOperationResponse) => {
  // custom code
  const response = (message as IServiceConnectionResponse).getResponseData();
  console.log(response.getStatus());
  console.log(response.getBody());
}).catch((error: IErrorData) => {
  // custom code
});
```

Here's a JavaScript example:

```
const frameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();

const restCallRequest =
  (frameworkProvider.requestHelper.createPublishRequest('InvokeServiceConnection'));
restCallRequest.setServiceConnectionId('interactions/update_interactions');
restCallRequest.setParameters({ "interactions_Id": "12345" });
restCallRequest.setBody({ "StatusCd": "ORA_SVC_CLOSED" });

globalContext.publish(restCallRequest).then((message) => {
  // custom code
  const response = message.getResponseData();
  console.log(response.getStatus());
  console.log(response.getBody());
}).catch((error) => {
  // custom code
});
```

FocusTab Operation

This operation focuses on a particular browser tab, or an MSI tab opened in the current browser tab. This is an operation specific to the tab, so it should be executed on TabContext. The response will fetch the details of the current tab's context, meaning the focused tab's context and the previous tab's context.

Note: FocusTab operation is publishable from the TabContext level.

Here's a TypeScript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const payload: IOperationRequest =
  frameworkProvider.requestHelper.createPublishRequest('cxEventBusFocusTabOperation');
tabContext.publish(payload).then((message: IOperationResponse) => {
  const currentTab: ITabContext = (message as
  IFocusTabResponseData).getResponseData().getCurrentTab();
  const previousTab: ITabContext = (message as
  IFocusTabResponseData).getResponseData().getPreviousTab();
}).catch((error: IErrorData) => {
  });
```

Here's a JavaScript example:

```
const publishFocusTab = async () => {

  const frameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
  const tabContext = await frameworkProvider.getTabContext();
  const payload =
  frameworkProvider.requestHelper.createPublishRequest('cxEventBusFocusTabOperation');
  tabContext.publish(payload).then((message) => {
    const currentTab = message.getResponseData().getCurrentTab();
    const previousTab = message.getResponseData().getPreviousTab();
  }).catch((error) => {
  });
};
```

CloseTab Operation

This operation is used to close a particular browser tab, or an MSI tab that's opened in the current browser tab. This is an operation specific to the tab and should be executed on TabContext. The response fetches the closed tab's identifier details in the ITabInfo object.

Note: CloseTab Operation is an operation publishable from the TabContext level.

Here's a TypeScript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
  CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const payload: IOperationRequest =
  frameworkProvider.requestHelper.createPublishRequest('cxEventBusCloseTabOperation');
tabContext.publish(payload).then((message: IOperationResponse) => {
  const tabInfo: ITabInfo = (message as ITabCloseOperationResponse).getResponseData();
  const browserTabId: string = tabInfo.getTabId();
  const MSITabId: string = tabInfo.getMSITabId();
}).catch((error: IErrorData) => {
```

```
});
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const payload =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusCloseTabOperation');
tabContext.publish(payload).then((response) => {
const tabInfo = response.getResponseData();
const browserTabId = tabInfo.getTabId();
const MSITabId = tabInfo.getMSITabId();
}).catch((error) => {
});
```

Pop Operation

This operation is used to open any page from an external application and perform all the UEF supported operations and subscriptions on the TabContext object, which they will get as the response of this operation.

The response of this operation will have the TabContext object, which is a reference object of the opened browser tab or MSI tab in which the new page is opened. This operation enables the user to open the page either in a browser tab or in MSI tab. UEF supports opening Service Request, Case, Contact, Account, Lead, Activity, and Application UI Pages.

There's three methods used to open pages.

To open a Service Request, Case, Contact, Account, Lead, and Activity in IPopFlowInAppRequest, set the RecordType of the page. It can be a Service request, Case, Contact, Account, Lead, and Activity. RecordType is the only required value. In IPopFlowRequest, you can mention whether you need to open that in a new browser tab or the same tab, and if any parameter must pass to the page,. If you need to open the edit page of these pages, Service Request, Case, Contact, Account, Lead, and Activity, set the record in in **recordId**.

To open the Application page, In IPopFlowAppUIRequest, Set ApplicationUI Name, Flow, and Page. In IPopFlowRequest,you can mention whether you need to open that in a new browser tab or in the same tab, and if any parameter needs to pass to the page.

To open any page, In IPopFlowGenericRequest, Set Flow and Page. In IPopFlowRequest , you can mention whether you need to open that in a new browser tab or the same tab and if any parameter needs to pass to the page.

Open ServiceRequest Create Page using Pop

Here's a TypeScript example of IPopFlowInAppRequest opening the Create Service Request page:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('ServiceRequest');
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
const tabContext: ITabContext = response.getResponseData();
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('ServiceRequest');
const response = await globalContext.publish(requestObject);
```



```
const tabContext = response.getResponseData();
```

Open ServiceRequest Edit Page using Pop

Here are examples of IPopFlowInAppRequest opening an Edit Service Request page with the detail view:

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('ServiceRequest');
requestObject.setRecordId('SR0000282245');
requestObject.setInputParameters({view: 'detail'});
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
const tabContext: ITabContext = response.getResponseData();
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('ServiceRequest');
requestObject.setRecordId('SR0000282245');
requestObject.setInputParameters({view: 'detail'});
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
const tabContext = response.getResponseData();
```

Pop Case Create Page

Here are examples of IPopFlowInAppRequest used to open and Create a Case page:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('Case');
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
const tabContext: ITabContext = response.getResponseData();
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('Case');
const response = await globalContext.publish(requestObject);
const tabContext = response.getResponseData();
```

Pop Case Edit Page

Here are examples of IPopFlowInAppRequest opening an Edit Case page in a new browser tab:

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
```

```
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('Case');
requestObject.setRecordId('CDRM2245');
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
const tabContext: ITabContext = response.getResponseData();
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('Case');
requestObject.setRecordId('CDRM2245');
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
const tabContext = response.getResponseData();
```

Pop Contact Create Page

Here are examples of IPopFlowInAppRequest used to open a Create Contact page.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('Contact');
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('Contact');
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
```

Pop Contact Edit Page

Here are examples of IPopFlowInAppRequest used to open an Edit Contact page.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('Contact');
requestObject.setRecordId('CDRM_123456');
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('Contact');
requestObject.setRecordId('CDRM_123456');
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
```

Pop Account Create Page

Here are examples of IPopFlowInAppRequest used to open a Create Account page.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('Account');
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('Account');
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
```

Pop Account Edit Page

Here are examples of IPopFlowInAppRequest used to open an Edit Account page.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('Account');
requestObject.setRecordId('CDRM_123456');
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('Account');
requestObject.setRecordId('CDRM_123456');
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
```

Pop Open AppUI Page

Here are examples of the `IPopFlowAppUIRequest` used to open 360 pages.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowAppUIRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowAppUIRequest;
requestObject.setApplicationUIName('advanced-customer-care');
requestObject.setFlow('main');
requestObject.setPage('main-start/main-dashboard');
requestObject.setOpenPageInNewBrowserTab(true);
requestObject.setInputParameters({ contactPartyNumber: "CDRM_943646", selectedAccountId: "9466225076" });
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setApplicationUIName('advanced-customer-care');
requestObject.setFlow('main');
requestObject.setPage('main-start/main-dashboard');
requestObject.setOpenPageInNewBrowserTab(true);
requestObject.setInputParameters({ contactPartyNumber: "CDRM_943646", selectedAccountId: "9466225076" });
const response = await globalContext.publish(requestObject);
```

Pop Open Generic Page

Here are examples of the `IPopFlowGenericRequest` to open the Article page.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowGenericRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowGenericRequest;
requestObject.setFlow('service/ec/container/sr');
requestObject.setPage('view-article');
requestObject.setInputParameters({ answerId: "10006003" });
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setFlow('service/ec/container/sr');
requestObject.setPage('view-article');
requestObject.setInputParameters({ answerId: "10006003" });
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
```

Pop Open Generic Page with Full URL

The following are examples for `IPopFlowGenericRequest` to open the Article page (full URL including https):

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowUrlRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowUrlRequest;
requestObject.setUrl('https://url');
requestObject.setOpenPageInNewBrowserTab(true);
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setUrl('https://url');
requestObject.setOpenPageInNewBrowserTab(true);
const response = await globalContext.publish(requestObject);
```

Pop Open New Object with Set Operation

The following shows examples of Pop with a Set operation.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('ServiceRequest');
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
const tabContext: ITabContext = response.getResponseData();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const setFieldRequestObject: ISetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation') as
ISetFieldValueOperationRequest);
setFieldRequestObject.field().setValue('ServiceRequest.Title', 'New Title');
setFieldRequestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');
recordContext.publish(setFieldRequestObject).then((message) => {
const setFieldResponse = message as ISetFieldValueResponse;
//custom code
}).catch((error: IErrorData) => {
// custom code
});
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('ServiceRequest');
const response = await globalContext.publish(requestObject);
const tabContext = response.getResponseData();
const recordContext = await tabContext.getActiveRecord();
const setFieldRequestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation');
setFieldRequestObject.field().setValue('ServiceRequest.Title', 'New Title');
setFieldRequestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');
recordContext.publish(setFieldRequestObject).then((message) => {
// custom code
}).catch((error) => {
// custom code
});
```

```
});
```

Pop Open Existing Object with Get Operation

The following shows examples of Pop with a Get operation.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext: IGlobalContext = await frameworkProvider.getGlobalContext();
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('ServiceRequest');
requestObject.setRecordId('SR0000282245');
const response: IPopFlowResponse = await globalContext.publish(requestObject) as IPopFlowResponse;
const tabContext: ITabContext = response.getResponseData();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const getFieldRequestObject: IGetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusGetFieldValueOperation') as
IGetFieldValueOperationRequest);
getFieldRequestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.publish(getFieldRequestObject).then((message) => {
const response = message as IGetFieldValueResponse;
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```

Here's a JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('ServiceRequest');
requestObject.setRecordId('SR0000282245');
const response = await globalContext.publish(requestObject);
const tabContext = response.getResponseData();
const recordContext = await tabContext.getActiveRecord();
const getFieldRequestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusGetFieldValueOperation');
getFieldRequestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);
recordContext.publish(getFieldRequestObject).then((response) => {
// custom code
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
}).catch((error) => {
// custom code
});
```

Pop Open New Object with a Set Required Field and Save Operation

The following shows examples of Pop with the Set operation and Save operation.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const openerTabContext: ITabContext = await frameworkProvider.getTabContext();
```

```
const requestObject: IPopFlowInAppRequest =
frameworkProvider.requestHelper.createPublishRequest('PopOperation') as IPopFlowInAppRequest;
requestObject.setRecordType('ServiceRequest');
const response: IPopFlowResponse = await openerTabContext.publish(requestObject) as IPopFlowResponse;
const tabContext: ITabContext = response.getResponseData();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
//set field value
const setFieldRequestObject: ISetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation') as
ISetFieldValueOperationRequest);
setFieldRequestObject.field().setValue('ServiceRequest.Title', 'New Title');
recordContext.publish(setFieldRequestObject).then((message) => {
const setFieldResponse = message as ISetFieldValueResponse;
//save operation
const saveRequestObject: IOperationRequest =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSaveRecordOperation');
recordContext.publish(saveRequestObject).then((message) => {
const response = message as ISaveRecordResponse;
const oldObjectId = response.getResponseData().getOldObjectId();
const newObjectId = response.getResponseData().getObjectId();
const objectType = response.getResponseData().getObjectType();
}).catch((error: IErrorData) => {
// custom code
});
}).catch((error: IErrorData) => {
// custom code
});
```

Here's a JavaScript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const globalContext = await frameworkProvider.getGlobalContext();
const requestObject = frameworkProvider.requestHelper.createPublishRequest('PopOperation');
requestObject.setRecordType('ServiceRequest');
const response = await globalContext.publish(requestObject);
const tabContext = response.getResponseData();
const recordContext = await tabContext.getActiveRecord();
//set field value
const setFieldRequestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation');
setFieldRequestObject.field().setValue('ServiceRequest.Title', 'New Title');
setFieldRequestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');
recordContext.publish(setFieldRequestObject).then((message) => {
//save operation
const saveRequestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSaveRecordOperation');
recordContext.publish(saveRequestObject).then((response) => {
const oldObjectId = response.getResponseData().getOldObjectId();
const newObjectId = response.getResponseData().getObjectId();
const objectType = response.getResponseData().getObjectType();
}).catch((error) => {
// custom code
});
}).catch((error) => {
// custom code
});
```

Get Field Value Operation

This operation is used to fetch the current value of a field of a particular Record or Object (such as, to get the current value of the Title field of SR). You can set multiple fields in the request object of this operation to fetch multiple fields in one getFieldValue operation. This is a record-specific operation.

The following code sample shows an example in TypeScript of publishing GetFieldValue operation where field names are passed.

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const requestObject: IGetFieldValueOperationRequest = (frameworkProvider.requestHelper.
createPublishRequest('cxEventBusGetFieldValueOperation') as IGetFieldValueOperationRequest);
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.publish(requestObject).then((message) => {
const response = message as IGetFieldValueResponse;
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```

The following code sample shows an example in JavaScript for publishing GetFieldValue operation:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();
const requestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusGetFieldValueOperation');
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);
recordContext.publish(requestObject).then((response) => {
// custom code
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
}).catch((error) => {
// custom code
});
```

Set Field Value Operation

This operation is to update the value of a particular field of a particular Record or Object (for instance, updating the Title field of SR). This is a record-level operation; so, you must call the publish API on top of the record context.

The following code sample shows an example in TypeScript of publishing SetFieldValue Operation where field names are passed.:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();

const requestObject: ISetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation') as
ISetFieldValueOperationRequest);
requestObject.field().setValue('ServiceRequest.Title', 'New Title');
requestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');

recordContext.publish(requestObject).then((message) => {
const response = message as ISetFieldValueResponse;
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```



```
});
```

The following code sample shows an example in JavaScript for publishing SetFieldValue operation:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();

const requestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation');
requestObject.field().setValue('ServiceRequest.Title', 'New Title');
requestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');

recordContext.publish(requestObject).then((message) => {
// custom code
}).catch((error) => {
// custom code
});
```

Save Record Operation

This operation is used to save an open Record or Object in the UI. (such as save an SR form from the Edit page or save an SR from Create page).

The response of SaveRecordOperation gives information about the oldObjectId and actual object identifier after saving. This response is similar to the response from OnAfterSave event subscription response. The old identifier and new identifier is different only when a save event happens on a new object. For example, while creating a new object UEF assigns a -ve identifier to it, and this -ve identifier is received in the ContextOpen event's subscription response. After saving, this object is assigned an actual identifier from the database.

Note: The SaveRecord Operation is an operation publishable from RecordContext level.

The following code sample shows an example in TypeScript for publishing SaveRecord operation:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const requestObject: IOperationRequest =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSaveRecordOperation');
recordContext.publish(requestObject).then((message) => {
const response = message as ISaveRecordResponse;
const oldObjectId = response.getResponseData().getOldObjectId();
const newObjectId = response.getResponseData().getObjectId();
const objectType = response.getResponseData().getObjectType();
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```

The following code sample shows an example in JavaScript for publishing SaveRecord operation:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();

const requestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSaveRecordOperation');
```

```
recordContext.publish(requestObject).then((response) => {  
  // custom code  
  const oldObjectId = response.getResponseData().getOldObjectId();  
  const newObjectId = response.getResponseData().getObjectId();  
  const objectType = response.getResponseData().getObjectType();  
}).catch((error) => {  
  // custom code  
});
```

Field Name Mapping

Field Name Mapping

There are certain events and operations in the UI Events Framework which accept field names of an object or record and perform certain operations or listen to certain events.

Field level operations and events such as the FieldValueChange event, the GetFieldValue operation, and the SetFieldValue operation require field names' details contained in an object or record. An object's field name details can be obtained using the following API:

```
# metadata of a specific <object>  
curl -X GET -u username:password https://servername.fa.us2.oraclecloud.com/crmRestApi/resources/11.13.18.05/  
<object>/describe
```

Here are some examples of API usage for ServiceRequest, Case and Contact objects:

```
# serviceRequest object  
curl -X GET -u username:password https://servername.fa.us2.oraclecloud.com/crmRestApi/resources/11.13.18.05/  
serviceRequests/describe  
  
# case object  
curl -X GET -u username:password https://servername.fa.us2.oraclecloud.com/crmRestApi/resources/11.13.18.05/  
cases/describe  
  
# contact object  
curl -X GET -u username:password https://servername.fa.us2.oraclecloud.com/crmRestApi/resources/11.13.18.05/  
contacts/describe
```

```
1  {
2    "Resources": {
3      "serviceRequests": { ← object Type
4        "discrColumnType": false,
5        "title": "Service Request",
6        "titlePlural": "Service Requests",
7        "group": "OEC",
8        "usage": "BusinessObject",
9        "attributes": [
10         {
11           "name": "Title", ← field MetaData
12           "type": "string",
13           "updatable": true,
14           "mandatory": true,
15           "inputRequired": true,
16           "queryable": true,
17           "allowChanges": "always",
18           "precision": 400,
19           "title": "Title",
20           "maxLength": "400",
21           "properties": {
22             "DISPLAYWIDTH": "40",
23             "FIELDORDER": "0.0"
24           },
25           "annotations": {
26             "description": "The brief title of the service request."
27           }
28         },
29         {
30           "name": "SeverityCd",
31           "type": "string",
32           "updatable": true,
33           "mandatory": false
```

Here the metadata of all objects:

```
# metadata of all the objects' names with field details
curl -X GET -u username:password https://servername.fa.us2.oraclecloud.com/crmRestApi/resources/11.13.18.05/
describe
```

All fields must be mapped using the following format:

```
ObjectType.FieldName
eg: ServiceRequest.Title
```

As an example, an order to subscribe or perform certain operations on SR Title form, one must pass the field name as ServiceRequest.Title in the event, operation request object.

Note: As part of the current release of ui-events-framework, we only support simple and LOV fields. Complex fields are supported in the coming releases.

Get Field Value operation

This operation is used to fetch the current value of a field of a particular Record or Object (for example, to get the current value of the Title field of SR). This is a record level operation and thus the user must call the Publish API on top of record context to perform this operation.

The following code sample shows an example in Typescript for publishing GetFieldValue operation where field names are passed.

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const requestObject: IGetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusGetFieldValueOperation') as
IGetFieldValueOperationRequest);
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.publish(requestObject).then((message) => {
const response = message as IGetFieldValueResponse;
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```

The following code sample shows an example in JavaScript for publishing GetFieldValue Operation where field names are passed.

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();
const requestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusGetFieldValueOperation');
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);
recordContext.publish(requestObject).then((response) => {
// custom code
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
}).catch((error) => {
// custom code
});
```

Set Field Value operation

This operation is used to update the value of a particular field of a particular record, object (For example, to update the Title field of SR). This is a record level operation and thus the user must call the publish API on top of record context to perform this operation.

The following code sample shows an example in Typescript for publishing SetFieldValue operation where field names are passed.

```
/// <reference path="uiEventsFramework.d.ts"/>
```

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();

const requestObject: ISetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation') as
ISetFieldValueOperationRequest);
requestObject.field().setValue('ServiceRequest.Title', 'New Title');
requestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');

recordContext.publish(requestObject).then((message) => {
const response = message as ISetFieldValueResponse;
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```

The following code sample shows an example in JavaScript for publishing SetFieldValue Operation where field names are passed.

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();

const requestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation');
requestObject.field().setValue('ServiceRequest.Title', 'New Title');
requestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');

recordContext.publish(requestObject).then((message) => {
// custom code
}).catch((error) => {
// custom code
});
```

Field Value Change Event

Here are examples of subscribing to FieldValueChange event where field names are passed.

Here's a Typescript example:

```
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();

const requestObject: IFieldValueChangeEventRequest =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusFieldValueChangeEvent') as
IFieldValueChangeEventRequest;
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.subscribe(requestObject, (message: IEventResponse) => {
const response = message as IFieldValueChangeEventResponse;
const fieldName: string = response.getResponseData().getFieldName();
const newValue: string | boolean | number = response.getResponseData().getNewValue();
const oldValue: string | boolean | number = response.getResponseData().getOldValue();
});
```

The following code sample shows an example in JavaScript example:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
```

```
const recordContext = await tabContext.getActiveRecord();
const requestObject =
frameworkProvider.requestHelper.createSubscriptionRequest('cxEventBusFieldValueChangeEvent')
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.subscribe(requestObject, (response) => {
const fieldName = response.getResponseData().getFieldName();
const newValue = response.getResponseData().getNewValue();
const oldValue = response.getResponseData().getOldValue();
});
```

Get Field Value Operation

This operation is used to fetch the current value of a field of a particular Record or Object (such as, to get the current value of the Title field of SR). You can set multiple fields in the request object of this operation to fetch multiple fields in one getFieldValue operation. This is a record-specific operation.

The following code sample shows an example in TypeScript of publishing GetFieldValue operation where field names are passed.

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'V1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();
const requestObject: IGetFieldValueOperationRequest = (frameworkProvider.requestHelper.
createPublishRequest('cxEventBusGetFieldValueOperation') as IGetFieldValueOperationRequest);
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);

recordContext.publish(requestObject).then((message) => {
const response = message as IGetFieldValueResponse;
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```

The following code sample shows an example in JavaScript for publishing GetFieldValue operation:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();
const requestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusGetFieldValueOperation');
requestObject.setFields(['ServiceRequest.Title', 'ServiceRequest.ProblemDescription']);
recordContext.publish(requestObject).then((response) => {
// custom code
const titleFieldValue = response.getResponseData().getField('ServiceRequest.Title').getValue();
const problemDescriptionValue =
response.getResponseData().getField('ServiceRequest.ProblemDescription').getValue();
}).catch((error) => {
// custom code
});
```

Set Field Value Operation

This operation is to update the value of a particular field of a particular Record or Object (for instance, updating the Title field of SR). This is a record-level operation; so, you must call the publish API on top of the record context.

The following code sample shows an example in TypeScript of publishing SetFieldValue Operation where field names are passed.:

```
/// <reference path="uiEventsFramework.d.ts"/>
const frameworkProvider: IUiEventsFrameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext: ITabContext = await frameworkProvider.getTabContext();
const recordContext: IRecordContext = await tabContext.getActiveRecord();

const requestObject: ISetFieldValueOperationRequest =
(frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation') as
ISetFieldValueOperationRequest);
requestObject.field().setValue('ServiceRequest.Title', 'New Title');
requestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');

recordContext.publish(requestObject).then((message) => {
const response = message as ISetFieldValueResponse;
// custom code
}).catch((error: IErrorData) => {
// custom code
});
```

The following code sample shows an example in JavaScript for publishing SetFieldValue operation:

```
const frameworkProvider = await
CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('MyFirstExtensionID', 'v1');
const tabContext = await frameworkProvider.getTabContext();
const recordContext = await tabContext.getActiveRecord();

const requestObject =
frameworkProvider.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation');
requestObject.field().setValue('ServiceRequest.Title', 'New Title');
requestObject.field().setValue('ServiceRequest.ProblemDescription', 'New Problem Description');

recordContext.publish(requestObject).then((message) => {
// custom code
}).catch((error) => {
// custom code
});
```


7 Catalog Management

How do I define a catalog for Service?

In most implementations, you must create a catalog of products and services. If you've already defined a catalog with the Base usage for Sales, it can be used in the Service offering.

You also 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've 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 correct granularity to 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. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Service Catalog
 - o Task: Manage Product Group Usage

The Manage Product Group Usage page appears.

2. Click the **Service** usage.
3. In the **Service: Details** section, on the **Product Groups** tab, add the root product group for the sales catalog.
4. 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 the Service usage.

To define a distinct catalog for the Service offering:

1. Define a product catalog specific to Service.
2. In the **Manage Product Group Usage** page, click the **Service** usage.
3. Associate the product catalog that you created to the Service usage, as described in the previous section.

Overview of Sales Catalogs

Using sales catalogs lets you:

- Use product group as a territory dimension so assignments can be made based on product.
- Give salespeople a mechanism to add product revenue to opportunities.
- Let salespeople 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 Cloud (Oracle CPQ). For information on the setup of individual products, see the topics on sales products.

Sales Catalog Key Features

Here are the key features of the sales catalog:

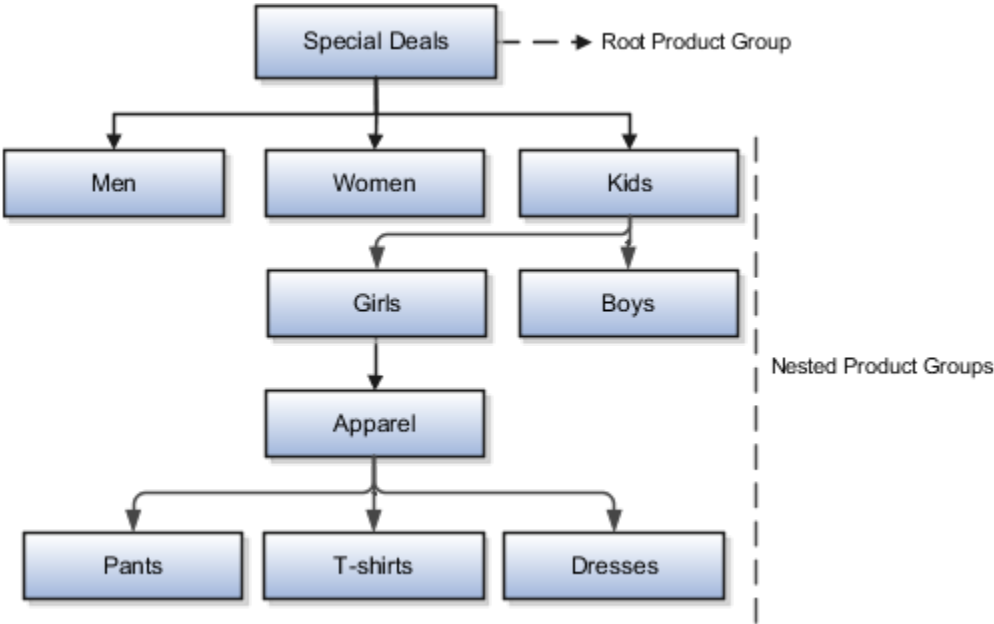
- Import your inventory list using standard import or import macros.
- Quickly build and deploy sales catalogs in a single administration UI.
- Build product groups in a hierarchy.
- Translate product group display name and description into different languages.
- Use the sales Products UI to create individual products that you add to the product catalog.
- Use price books and the default pricing region in the Products UI to price "standalone" products outside of the catalog 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.



Related Topics

- [Sales Catalog High-Level Setup Steps](#)
- [Add Products to the Catalog](#)
- [Validate the Sales Catalog](#)
- [Run the Refresh Denormalized Product Catalog Table Process](#)
- [Example of Creating a Sales Catalog](#)

Sales Catalog High-Level Setup Steps

If you didn't use the Setup Assistant to get started creating your catalog, follow the steps here to create the catalog.

This table shows the high-level setup steps and where to find more information about the step:

Note: If you're using the Setup Assistant to create your catalog, start with [Overview of Sales Catalog Setup](#). When you use the Setup Assistant, many steps are done for you by the assistant.

Description	Where to Find More Information
Step 1 Ensure that your prerequisite structures are set up before starting to create the catalog.	Prerequisite Setups for Sales Products
Step 2	Create the Root Product Group

Description	Where to Find More Information
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.	
Step 3 Add additional product groups to create the catalog hierarchy of product groups and subgroups. Once your hierarchy is set, you can add the product groups manually in the product groups pages, or you can import them. Importing products lets you use quick import macros or Import Management to create the entire product group and product hierarchy all at once. Note: If you're not integrating with an order management application or doing quoting, you can simply use a product group hierarchy without products. There's no need to use individual products unless you need to for downstream applications.	• Create the Product Group Hierarchy • Options for Importing Data Into Your Application
Step 4 If you're going to use individual products in the catalog, add products to the application. You can create products using the sales Products UI or the product model available with Oracle Supply Chain Management Cloud. You can also import products and the product hierarchy all at once. Or, you can use Import Management to import products and the hierarchy.	• If using the sales Products UI, see Create Products Manually in the UI. • If using Oracle Supply Chain Management (SCM) Cloud product model see these guides: ○ Oracle SCM Cloud Implementing Product Management ○ Oracle SCM Cloud Using Product Master Data Management • Options for Importing Data Into Your Application
Step 5 If you're going to use individual products in the catalog, add products to the product group hierarchy.	Add Products to the Catalog
Step 6 Publish the product group hierarchy that makes up the sales catalog. 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.	• Publish the Sales Catalog • Run the Refresh Denormalized Product Catalog Table

Description	Where to Find More Information
Step 7 To enable a sales catalog for use in the applications, you associate it with a "usage" called the Base usage.	Enable the Sales Catalog by Setting the Base Usage
Step 8 If 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 don't run the process, your product group hierarchy might not appear in the consuming applications.	Run the Refresh Denormalized Product Catalog Table
Step 11 If you've set up the browse catalog feature (needed only for classic Sales only), configure search and browse options.	Set Sales Catalog Search and Browse Options
Step 12 After you've published and enabled your catalog and, optionally, set up browsing, validate your setups.	Validate the Sales Catalog
Step 13 You can implement eligibility rules that enable salespeople to check product eligibility in opportunities.	Set Up Product Eligibility
Step 14 You can 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.	Territory Filtering of Products in the Catalog
Step 15 You can use the prebuilt Fusion Sales-Oracle Configure, Price, and Quote (Oracle CPQ) integration to let salespeople manage a variety of quote-related activities.	These guides: • How do I get started with the Oracle Revenue Transformation Solution? • Integrating Fusion Sales with Oracle CPQ guide

Add Products to the Catalog

Now that you have created products following the steps in the Products chapter of this guide, or, if you have imported them, your next step is to add them to the product groups that make up the sales catalog hierarchy.

Here's how to manually add products to the sales catalog product group hierarchy.

1. In Setup and Maintenance, go to the **Sales offering > Sales Catalog and Products task > Manage Product Groups task**.
2. In the Manage Product Groups page, in the product group hierarchy, select the product group that you want to add products to.
3. Lock the product group for editing by clicking the **Lock** button.
4. Click the **Products** tab for the product group you selected.
5. In the **View** filter, ensure that the Administration view is selected.
6. In the products table, select **Actions > Select and Add**.
7. In the Select and Add: Products screen, search for and select the product you're adding.
8. 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.

9. Click the **Publish** button to publish the product group.
10. Finally, click **Yes** in the Confirm Publish dialog window and then dismiss the confirmation message. The application publishes the product group.
11. Save your changes.

To select the organization that provides the set of products available for marketing and sales activities, select an option in the QSC_SALES_PRODUCTS_INVENTORY_ORG_ID profile value.

For more information about how to create products manually in the sales UI, see the Products chapter in this guide. To understand how to import products and the product hierarchy, see the [product import topics](#).

Of course, if you're integrating with Oracle Supply Chain Management (SCM) Cloud, you may be creating product in that application's Products screens.

Related Topics

- [Overview of Sales Products](#)
- [What are my data import options?](#)

Validate the Service Catalog

Here's how you validate that the product groups are displayed on your SRs after you have published and enabled your catalog:

1. Sign into Service Center 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

- [Sales Catalog High-Level Setup Steps](#)
- [Run the Refresh Denormalized Product Catalog Table Process](#)
- [Best Practices for Sales Catalog Setup](#)

8 Configure Service Request Outcomes and Resolutions

Outcomes and Resolutions

Oracle Fusion Service provides ready-to-use outcomes and resolutions. Administrators can configure additional outcomes and resolutions.

Oracle recommends that you use the task if you want to associate the outcomes, and display the configured items as drop-down list menu in the Resolve Service Request dialog box.

The **Manage Service Request Outcomes and Resolutions** task is used when administrators want to create new resolutions and map them to outcomes. The task must only be used when administrators want to create new resolutions and don't want to map them to outcomes

Manage Outcomes and Resolutions

The Manage Service Request Outcomes and Resolutions tasks are used when administrators want to create new outcomes, resolutions, and map resolutions to outcomes.

Follow these steps to configure Outcomes and Resolutions, and map them:

Step 1: Configure Outcomes

To add an outcome:

1. Sign in to the application as an administrator.
2. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Service Request
 - o Task: Manage Service Request Outcomes and Resolutions.
3. In the **Manage Service Request Outcomes and Resolutions** page, select the **Outcome** tab.
4. Click the **Add** icon to add a new outcome.
5. In the new row that displays, enter the fields such as lookup code, display sequence, meaning, description, and so on to correspond to your needs.

Note: Oracle recommends that you name the outcome lookup code starting with SVC_ to differentiate between customer-created outcomes and ready-to-use outcomes.

6. Click **Save and Continue**.

If you have selected the **Enabled** check-box, the outcome is added to the **Outcomes** list under the **Map Resolutions to Outcomes** section, else the new outcome is only listed in the **Outcomes** list.

Step 2: Configure Resolutions.

Administrators can configure new resolutions in the **Manage Service Request Outcomes and Resolutions** task.

To add resolutions:

1. In the **Manage Service Request Outcomes and Resolutions** task page, select the **Resolution** tab.
2. Click the **Add** icon to add a new resolution.
3. In the new row that displays, enter the fields such as lookup code, display sequence, meaning, description, and so on to correspond to your needs.
4. Click **Save and Continue**.

If you have selected the **Enabled** check-box, The resolution is added to the **Available Resolutions** list under the **Map Resolutions to Outcomes** section.

Step 3: Map Outcomes to Resolutions.

Administrators can map the outcomes to the available resolutions in the **Map Resolutions to Outcomes** section in the page.

To manage the outcomes and resolutions:

1. In the **Map Resolutions to Outcomes** section, do the following to map the outcomes to resolutions.
 - o Select an outcome in the **Outcomes** list.
 - o Select the resolutions from **Available Resolutions** list and move them to the **Selected Resolutions** list.
 - o To remove an existing mapping, select the resolutions from the **Selected Resolutions** list and move them to the **Available Resolutions** list.
2. Click **Save and Close**.

The outcomes are now mapped to the resolutions.

Once the outcomes and resolutions have been configured, these appear in the **Resolve Service Request** dialog box in the **Outcome and Resolution Code** list, and the user can enter simple and consistent resolution information about the SR.

Ready to Use Outcomes and Resolutions

The following table lists the ready to use outcomes and resolutions.

Outcomes	Resolutions
Solution Provided	Knowledge Article FAQ Workaround identified
Unable to Solve Defect Logged	Enhancement Logged Training Required No Fault Found

Outcomes	Resolutions
Customer Abandoned	Failed attempts to contact Resolved by Customer

Manage Resolutions

The administrator uses the Manage Service Request Resolutions to add resolutions only when they don't want to associate them to outcomes.

To add, edit, or delete resolutions without associating them with outcomes:

1. Sign in to the application as an administrator.
2. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Service Request
 - o Task: Manage Service Request Resolutions.
3. In the Manage Service Request Resolutions page, all resolutions for the lookup type **ORA_SVC_SR_RESOLUTION_CD** are displayed.
4. Click **Add** to add an outcome under the relevant lookup type. You also modify or delete the outcomes here.
5. Click **Save and Close**.

9 Work Assignment and Routing

Overview of Assignments, Mappings, and Rules

For more information about assignment mappings, assignment rules, and configuring assignment, see the following topics in the Oracle Fusion Cloud Sales Automation Implementing Sales guide:

Topic in the Oracle Fusion Cloud Sales Automation Implementing Sales guide	What the topic covers
Considerations for Configuring Assignment	• Phases of assignment • Setup tasks used for an assignment configuration • Assignment objects • Attributes • Mapping sets and mappings • Rule categories, rule sets, and rules
Assignment Mapping Set Components	How the mapping set components work together in assignment processing.
Examples of Creating Assignment Mappings	How to create different types of assignment mappings: attribute, dimension, and literal.
Assignment Rule Components	How the following assignment rules components work together: rule categories, rule sets, and rules.
Examples of Creating Assignment Rules	How to create assignment rules using rule sets, rules, conditions, and actions.

Related Topics

- [Considerations for Configuring Assignment](#)
- [Assignment Mapping Set Components](#)
- [Examples of Creating Assignment Mappings](#)
- [Assignment Rule Components](#)
- [What types of assignment rules can I create?](#)

Implementation Concepts for Work Assignment

What are work assignments?

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 use rule-based assignment to assign additional resources to objects.

Candidate and Work Objects

When setting up assignments, 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, .
- Candidate objects are the possible pool of assignment candidates, for example, resources.

Rule-Based Assignment

Rule-based assignment lets you set up more 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 use rules to assign a certain agent to a certain queue when the customer is 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 enable you to further configure the application behavior.

Scheduled Processes

Scheduled processes are batch jobs that capture data and let business objects to act on that data. You must schedule several processes when using assignment. See the *Service Request Queue Assignment* topic in the Oracle Fusion Cloud Sales Understanding Scheduled Processes guide.

Assignment Reports

You 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, see the following resources:

- 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.

Assignment Mappings

How do I export and import assignment objects and rules setup data?

You can understand how to export and import assignment objects and rules setup data, along with the points to consider while moving the setup data in this topic.

Most 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 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 and 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 decides 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 compressed 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.

Points to Consider to Export and Import Setup Data

Based on your implementation, you can follow different approaches while exporting and importing assignment setup data.

Consider the following points:

- Because your implementation is using rule-based assignment, the implementation project must include both the Assignment Objects and Assignment Rules setup tasks.
- 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 isn't 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, don't delete assignment rule conditions. Instead, set the rule to inactive and then recreate the rule excluding the condition that's no longer needed.

Related Topics

- [How do I export and import assignment objects and rules setup data?](#)

Assignment Rules

How You Define Queue Assignment Rules

You can use assignment rules to automatically assign to queues when they are created or updated. And these assignment rules can be run on a schedule.

are treated as work objects and queues are treated as candidate objects. You can define your rules to select the best candidate (queue) for each work object .

Note: assignment rules are defined using rule-based assignment. Territory-based assignment doesn't apply to .

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

- The attributes of queues that you want to use as criteria for your rule assignments.
- The attributes of that 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 assignment objects.

Related Topics

- [Assignment Rule Components](#)
- [Update Existing Setup Data](#)
- [Create a Queue](#)
- [What happens when I change queue properties?](#)

How do I manage assignment objects?

You need to select attributes from the service request (SR) assignment object that you want to make available in your rules. But note that this procedure is optional, because ready-to-use fields are already provided for all the objects.

Note: You must not change the configuration for any of the ready-to-use objects. If you change the configuration, the Service Request Queue Assignment process can't run successfully. But you can add new attributes or new child entities for the objects.

(Optional) To manage SR assignment objects:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Communication Channels
 - Task: Manage Service Assignment Objects
2. On the Manage Service Assignment Objects page, add the queue attributes that you want to be available when you're setting up your rules:
 - a. Click **Queue**.
 - b. Click the **Attributes** tab.
 - c. Add attributes to the list by clicking the **Plus** icon, and then selecting the **View Object Attribute** from the drop-down list.
 - d. Click **Save**.
3. Add the service request attributes that 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, and then selecting the **View Object Attribute** from the drop-down list.
 - d. Click **Save**.

How do I manage assignment rules?

Let's go over how you define the rules for service assignment.

Here are some guidelines that you need to follow when you're defining matching rules for assigning work items to queues:

- You must define the rule set with Number of Candidates = 1. The application enables only one queue to be assigned to a service request (SR).
- You've 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 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 didn't 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.

You can use the operators listed in this table to define the conditions for SR assignment rules.

Operator	Description	Value	Used in hierarchy?	Example
Equals	The value of a field equals a specific value.	Single	No	If the rule condition is set as <i>Service Request Severity Equals High</i> , only an SR with the severity value of High matches the condition.
Does not equal	The value of a field doesn't equal a specific value.	Single	No	If the rule condition is set as <i>Service Request Severity Does not equal High</i> , an SR with severity value of Low or Medium matches the condition.
In	The value of a field is one of a list of values.	Single or Multiple	No	If the rule condition is set as <i>Service Request Problem type Code In Docs, Product</i> , an SR with the problem type code value of Docs or Product matches the condition.
In including children	The value of a field is one of a list of values. Indicates that the rule applies if an attribute value matches to any of the attributes in the parent-child hierarchy of the current attribute.	Single or Multiple	Yes	If the rule condition is set as <i>Service Request Product Group In including children Apple</i> , any SR 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.
Not in including children	The value of a field isn't in a list of values. This value is only relevant for category or product fields that are hierarchical.	Single or Multiple	Yes	If the rule condition is set as <i>Service Request Product Group Not in including children iPad</i> , an SR with the Product Group value of iPad or any of its children, such as iPad1 or

Operator	Description	Value	Used in hierarchy?	Example
	Indicates that the rule applies if the specified attribute value matches the top level of the attribute. This option doesn't include the attribute values of the children of the current attribute.			iPad2 doesn't match the condition. However, an SR with the Product Group value of Apple, iPhone, or iPhone 6s matches the condition.
Not in	The value of a field isn't in a list of values.	Single or Multiple	No	If the rule condition is set as <i>Service Request Problem Type Code Not in Docs, Product</i> , an SR with the problem type code value of User doesn't match the condition. However, an SR with a problem type code of Docs or Product is a match.
Is blank	The value of a field doesn't contain a value.	N/A	No	If the rule condition is set as a <i>Service Request Internal Priority Code Is blank</i> , an SR without an internal priority code value is a match.
Is not blank	The value of a field contains any value.	N/A	No	If the rule condition is set as <i>Service Request Account Is not blank</i> , an SR with an associated account is match, but not an SR without an account.

To manage SR assignment rules:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Service Assignment Rules
2. On the Manage Service Assignment Rules page, select a **Category** based on the following.
 - o **Generic Queuing Rules:** Rules set for all types of work items.
 - o **HR Service Request Queuing Rules:** Rules set for HR Help Desk SR.
 - o **Service Request Queuing Rules:** Rules set for CRM SR.
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.
5. In the **Create Rule** window, 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 the list of operators.

8. (Optional) Add more conditions.
9. In the Action: Assign Queue region, 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: You must republish the assignment rules each time the rule is changed. Also republish the rules each time the associated queue is deleted, enabled, or disabled.

How do I set queue assignment rules?

You can create assignment rules by using rule sets, rules, conditions, and actions.

The assignment engine uses your rules to evaluate and recommend candidate assignments for specified work objects.

Note: A particular work item can be assigned only to a single queue. So it's good to be careful while defining rules. You can't define different rules that assign the same work item to different queues.

To create a service assignment rule:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Service Assignment Rules
2. On the Manage Service Assignment Rules page, select from the **Category** drop-down list:
3. Do one of the following:
 - o Create a rule set by clicking the **Plus** icon and specifying the required values.
 - o Select an existing rule set.
4. Create a rule within the rule set by completing the following steps:

- a. Click the **Plus** icon in the **Rule set: Rules** region.
- b. On the Create Rule page, specify a rule name.
- c. (Optional) Specify a description, an effective start date, and an end date for the rule.

If you don't select an effective start date and end date, the rule comes into immediate effect and lasts indefinitely.

- d. (Optional) Select the Inactive option if you want to enable the rule at a later date.

Note: You can create multiple rules within a rule set.

5. Specify an assignment condition as described later in this topic, in the "Specify a Condition and an Action" section.

This assignment condition is evaluated before a rule-based assignment is made.

6. Select a queue to which is assigned if the condition is met.

Specify a Condition and an Action

To specify a condition and an action:

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 the **Plus** icon to add a condition.

Note: If you don't add a condition, all your work items are assigned to the queue that you add in step 4.

3. Select the object, an attribute, an operator for the condition, and a value.
4. Add an action to be performed when the condition is satisfied. Click the **Plus** icon to select and add a queue.

CAUTION: Oracle recommends that you always add an action. If you don't add an action, the assignment engine acts unpredictably and you may face issues in the assignment of work items to queues.

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**.

Related Topics

- [How do I update existing setup data?](#)

Tips for Creating Assignment Rules and Rule Sets

Summary of Tips for Creating Assignment Rules and Rule Sets

To maximize the effectiveness of your assignment rules, Oracle recommends that you follow these tips while creating them. Each of them is described in detail in separate topics:

- Add a condition and an action for every rule.
- Create unique rules.
- Create minimum number of rule sets.
- Publish rules after you rename them.
- Set incomplete rules and rule sets to inactive.

Add a Condition and an Action for Every Rule

It's important to remember that you always add a condition and an action for every rule. Any rule without a condition or action can lead to unpredictable results.

- If you don't add a condition, any work item will match this rule. This type of rule will conflict with other well-defined rules. This situation results in a work item matching multiple rules, so it's randomly assigned to any queue that matches any of these rules.
- Add an action to be performed when the condition is satisfied. You add a queue as part of the action. If you don't add an action, the Assignment Manager acts unpredictably and you may face issues in the assignment of work items to queues.

Note: Some examples of incomplete rules that you must avoid:

- If no condition, then assign the work item to the Q1 queue.
- If severity of the work item is high, then you don't specify any queue.

For more information, see [Set Rules for Queue Assignment](#).

Related Topics

- [How do I set queue assignment rules?](#)

How do I create unique rules?

Here are some pointers for creating unique rules:

- You must create unique rules in such a way that two different rules don't match the same work item. If multiple rules match the same work item, the Assignment Manager randomly picks one of those matching rules.
- You must also ensure that you don't have duplicate or identical rules. Duplicate or identical rules doesn't mean identical conditions. It means that multiple rules match the same incoming work item, even if the rules are defined on different fields or a combination of fields.
- If you can't create unique rules, you must put the rules in the same rule set and use scoring to identify which queue is picked by the Assignment Manager. You've 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 also set Filter as Top X and Number of candidates as 1. All the rules in a rule set are executed, and the queue with the highest total score is selected.

For more information, see [Manage Service Assignment Rules](#).

Note: Scoring applies only inside a rule set. But you might have scenarios where identical rules are present in different rule sets, and multiple rules from different rule sets match a work item. In such cases, the Assignment Manager randomly picks one of those identical rules.

Let's see a few examples of how you can make a rule unique:

Example 1 for Making a Rule Unique

Let's say you write the following rules:

- **Rule 1:** Variable A=1 and Variable B=2 (Action: Assign work item to Q1)
- **Rule 2:** Variable A=1, Variable B=2, and Variable C=3 (Action: Assign work item to Q2)

Then the Variable A=1 and Variable B=2 part is common in both the rules. When an incoming work item satisfies Rule 2, it also satisfies Rule 1. And the Assignment Manager randomly picks one of the two.

To make these two rules unique, you can add a condition about variable C in Rule 1. For example, you can add the following condition to Rule 1: C isn't equal to 3.

Note: Rule 1 and Rule 2 are identical if you get a work item with all 3 variables and both rules are matching.

Example 2 for Making a Rule Unique

Let's consider the following example involving different rule sets. In Rule Set 1, you've a rule with the condition C1=A. In Rule Set 2, you've a rule with the condition C2=B. Here, C1 and C2 are different fields.

Now let's say you've a work item where C1=A and C2=B. These two rules aren't unique, and they both match the work item. In such a scenario, any one of the rules is randomly picked, and a queue is selected. But suppose you want the work item to be assigned only to a queue called Q1 because it has C1, and you want C2 to be ignored. Then you must make the rules unique by setting conditions such as the following:

- **Rule 1:** If C1 equals A and C2 isn't equal to B, then assign to Q1.
- **Rule 2:** If C1 equals A and C2 equals B, then assign to Q2.

Note: If the matching rules are in different rule sets, you can't apply scoring. If the matching rules were in the same rule set, scoring would help. For example: You set the scoring of Rule 1 to 10 and Rule 2 to 5.

Create Minimum Number of Rule Sets

The Assignment Manager is designed to return only one queue for every work item.

If two rules from different rule sets match a work item, then the Assignment Manager randomly picks between the rules. So the work item may not be assigned to the queue that you would want. To avoid such situations, it's good to have only one rule set if possible. And you can apply scoring for the rules if required. That way, only one rule matches a work item. Regardless of the company size, Oracle recommends deploying only one rule set.

Note: You can still have multiple rule sets if you ensure that the rules within these rule sets are all unique. And within each rule set, you can have scoring for the rules. But if you have the same or similar rule in different rule sets, two rules from different rule sets can match the same work item.

Here's what the Assignment Manager does when you have multiple rule sets:

1. The Assignment Manager checks all rule sets. In one round of assignment, it checks all the rules under Rule Set 1, all the rules under Rule Set 2, and so on.

2. It finds some matching rules. Let's say the matching rules are Q1 and Q2 under Rule Set 1, and Q3 under Rule Set 2.
3. It randomly picks a rule among Q1, Q2, and Q3.

Note: You must not create more than the maximum limit of 500 rule sets. The Assignment Manager selects only 500 rule sets alphabetically to process them. But within every rule set, you can have any number of rules. An upper limit doesn't apply to the number of rules.

Publish Rules After You Rename Them

Oracle recommends that whenever you rename a rule, you must publish it immediately.

Let's say you create or rename a rule, and you don't publish it immediately. Sometimes, another administrator may delete it or create another rule with the same name. This causes confusion for the Assignment Manager while publishing. For the same reason, you must also avoid renaming a rule and deleting a rule in the same operation. You must publish the rule after every individual operation.

Set Incomplete Rules and Rule Sets to Inactive

You must set any incomplete rules or rule sets to inactive. While creating a rule or rule set, you must set the initial status to inactive.

This way, you avoid the risk of incomplete rules being published. You must first complete adding all the conditions and actions to a rule. After that's done and you're confident that the queue works as required, you can set it to active and publish it.

Here's why you must be careful. Suppose you created a rule and saved it, but you couldn't complete the rule for some reason. So you may not have added a condition and action to determine the queue for assignment. And let's say another administrator publishes the rules. Whenever any administrator publishes the rules, all active rules and rule sets that are saved until then are published, irrespective of who has created them. Incomplete rules that are set to active status are also published. When this happens, the Assignment Manager may pick a rule other than what you would prefer.

Work Assignments

How do I set 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 by using the Manage Capacities page.

Capacity is calculated separately for real-time work items such as SRs. For example: If the global capacity for open SRs is 30, an agent who's working on 30 open requests is considered to be 100 percent occupied.

If an agent who's accepting chat offers changes their presence state from present or available to busy, then the agent's chat availability status is automatically changed to unavailable, and no chats will be sent to that agent. For non real-time

work, agents can remain available, though. For instance, an agent might go to lunch and change presence state to busy which prevents real-time work offers, but the agent could still get non real-time work assigned (assuming the agent has capacity).

If a queue has multiple agents, the 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. .

Agent Presence

Presence indicates whether an agent who has signed in to the application is interacting with the application. Agents set their presence through the **Settings and Action** menu.

An agent can be online, but may not be available to handle work items. If an agent isn't signed in, then the presence is set to offline.

To configure the presence toolbar, see the Configure Presence and Availability Privilege topic in this chapter.

How do I set agent availability?

Agent availability indicates whether an agent is available. Agents are automatically assigned SRs. They don't need to make themselves available to handle an interaction or a work item. After an agent signs in to the application, they can set their availability to handle an interaction.

Agents can specify their availability for real-time and non-real-time work items. This means that agents can specify if they're available for chat, for SR, or for both, by using the icons on the Omnichannel toolbar.

Setting Presence

If an agent's 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, the agent's presence is automatically set to **Available for chat interactions**.

If an agent who's accepting chat offers changes their presence state from present or available to busy, then the agent's chat availability status is automatically changed to unavailable, and no chats will be sent to that agent. For non-real-time work, agents can remain available, though. For instance, an agent might go to lunch and change presence state to busy which prevents real-time work offers, but the agent could still get non-real-time work assigned (assuming the agent has capacity).

Note: Based on the SVC_OMNI_DISABLE_WORK_ITEM_AVAILABILITY_SELECTION profile option setting, the application restricts agents from blocking new work assignments. The application assigns new work to agents until their maximum capacity is reached.

How do I enable agents to set global presence across Fusion Service?

You set the profile option `ORA_SVC_MCA_GLOBAL_PRESENCE_ENABLED`.

Only enable this profile option if you're using Service Center in the Redwood user experience.

1. Navigate to Setup and Maintenance and open the Tasks panel tab.
2. Click **Search** from the list of displayed tasks.
3. Type Manage Admin in the **Search** field.
4. From the displayed list of tasks, click **Manage Administrator Profile Values**.
5. Search for the profile option code `ORA_SVC_MCA_GLOBAL_PRESENCE_ENABLED`.
6. In the Profile Values region, specify the values in the **Profile Value** column for the **Site Profile Level** to **Yes** if you want all users to set their presence from the global header.
7. Or, set the **Site level value** to **No** and add a new row for every user who would set their presence from the global header.
8. Click **Save and Close**.

Severity and Age of Requests

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.

The age of is also a factor while assigning work. A that's waiting in the queue for a longer period of time is given higher priority.

Related Topics

- [Add Severity Values to Requests](#)
- [Prevent Starvation of Low Priority Work Requests](#)

How do I create queues?

If you've enabled Omnichannel, there's a ready-to-use queue default queue. This topic discusses how to create queues and how default queues work.

Oracle recommends that the Default queue Distribution field is set to Automatic when Omnichannel is enabled. Available agents must also be associated with the Default queue.

Be aware that the Default queue contains work items that couldn't be assigned to any other queue because they don't match any of the defined assignment rules. You can add agents and teams to the Default queue and also change the queue properties, but you can't disable or delete this queue.

Depending on your requirements, you might want to create specific queues to route chats to a group of skilled agents. To do this, you can create more queues, staff them with agents, and create your routing rules to assign chats to that queue. Here are the steps for creating queues:

1. Navigate to **Service > Queues**.
2. Enter a name for the queue.
3. Check the **Activation** field.
4. Set Distribution to **Automatic (Push)**.
5. Click **Save and Continue**.
6. Click the **Resources** tab.
7. Select agents to add to the queue.
8. Click **Apply** when finished.
9. Click **OK**.

Note:

- The ready-to-use Default queue acts as a catch-all queue for Service Requests and Chats when assignment rules fail to evaluate matching conditions.

Here's what Oracle recommends:

- If you enable chat, you must set the Default queue to Automatic and assign agents to the queue. When the Default queue is automatic, all those chats that don't match any rules or they match a manual queue are assigned to the Default queue. If you set the Default queue to manual, such chats aren't routed to any agent, and no one would respond to those chats.

How do I assign work to agents?

Work assignment refers to interactions or work items such as service requests (SRs) 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 whether the associated queue is automatic or manual. For more information about the types of queues, see *"How to Assign Agents to Work Items"*.

Note: In automatic queues, the priority of SRs is calculated based on severity and wait time. SRs with highest severity are assigned first. If there are multiple SRs with the same severity, those SRs that have been in the queue for the longest time are assigned first. But this rule applies only to bigger volumes of SRs and many agents and not on a small scale. For more information, see *"Order in Which SRs Are Assigned"*.

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.

Here are the various ways an SR can be assigned to an agent:

- Manually set the **Assigned To** field in the Summary tab of the Service Request Details page.
- Click **Assign to Me** from the **Actions** menu on the Service Request Details page.
- Enable Omnichannel.

In what order are requests are assigned?

Here's how service requests (SRs) in an automatic queue are assigned to agents.

The application calculates the priority of SRs based on severity and waiting time. SRs with the highest severity are assigned first. If there are multiple SRs with the same severity, SRs that have been in the queue for the longest time are assigned first.

Note: It's possible that you may see exceptions to this rule when you're testing with a small number of agents. That's because the application is designed to work with millions of SRs and thousands of agents per day. And to do that, it has massive parallel processing power.

There are multiple servers in your production environment. Each server tries to assign SRs, competing with other servers to assign all the work among all available agents. SRs are assigned on multiple servers in parallel, otherwise the assignment tends to lag behind the incoming work.

Suppose your production environment has 15 servers. The application would act as if it's assigning SRs in batches of 15. The SRs from 1 to 15 would be assigned before the SRs 16 to 30. But the order within each batch isn't guaranteed. The first server that finds an available agent makes the assignment.

How do I assign work to a queue?

Here are the various ways in which you can assign a service request (SR) to a queue:

- Open an SR. From the **Queue** drop-down list, manually select the queue to which you want to assign the SR.
- Enable Omnichannel, which runs the queue assignment rules when the SR is created or when the queue is deleted from the SR.
- Click **Run Queue Assignment** from the **Actions** menu, which runs the queue assignment rules.
- Enable the Assign Service Request to Queue on Create profile option, which runs the queue assignment rules when the SR is created. The process is described below.
- Schedule the Queue Assignment job that runs the queue assignment periodically and processes all the open SRs that aren't assigned to a queue.

The preferred way of assigning an SR to a queue is by using Omnichannel assignment. Enabling Omnichannel has the following advantages:

- Enabling Omnichannel provides more features than queue assignment. It also enables the automatic assignment of agents. But if your business only requires queue assignment, you must carefully evaluate whether you need to enable Omnichannel.
- Only if you enable Omnichannel, the SR is assigned to the Default queue when the Assignment Manager rules haven't been defined or when the selected SR doesn't satisfy any active rule.
- When Omnichannel is enabled, automatic assignment of SRs to agents is always enabled on the Default queue.

Suppose you're assigning SRs to an automatic queue either manually or automatically by using assignment rules or Groovy scripts. If you enable Omnichannel, Oracle recommends that you also assign active agents for the automatic queue, so that SRs can be assigned to the agents.

Note: If you don't assign agents for the automatic queues, SRs assigned to those queues could grow exponentially over time. And this causes the application to slow down.

Enable the Assign Service Request To Queue On Create Profile Option to Assign an SR to a Queue

Here's how you enable the Assign Service Request To Queue On Create profile option:

1. Sign in to the application as an administrator.
2. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Service Request
 - Task: Manage Service Request Profile Options

Note: Or, you can open the Tasks panel tab and search for the **Manage Service Request Profile Options** task.

3. On the Manage Service Request Profile Options page, click **SVC_ASSIGN_TO_QUEUE_ON_CREATE**.
4. Navigate to the SVC_ASSIGN_TO_QUEUE_ON_CREATE: Profile Values region of the page.
5. In the **Profile Value** drop-down list for the Site **Profile Level**, select **Yes**.
6. Click **Save and Close**.

Note: If Omnichannel is enabled, this profile option is ignored.

Add Severity Values to Requests

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

As an administrator, you can set the severity values in the application.

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service

- Functional Area: Service Request
 - Task: Manage Service Request Severity Values
2. The `ORA_SVC_SR_SEVERITY_CD` lookup type contains the lookup codes. 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.
- 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 doesn't affect the actual severity of an SR . The actual severity is determined by the **Severity Ranking** field.
- **Severity Ranking:** Enter a value that depicts the priority level. An SR severity with a lower severity ranking is considered as having higher priority when they 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. SRs with disabled severity aren't displayed on the Service Request Edit page.
 - **Start Date** and **End Date:** Specify a date range within which the severity value is valid. If you don't 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's 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.
3. Add more lookup codes as required.
4. After you finish, click **Save and Close**.

Related Topics

- [How do I assign work to agents?](#)
- [How do I update existing setup data?](#)

Omnichannel

How do I automatically route chat work assignments to agents?

Routing rules identify queue assignments, which could be sent to a queue with live agents or the Oracle Digital Assistant (ODA) queue. They also escalate a chat from ODA to a live agent queue.

Note: Chat assignments must be created in the Generic Queuing Rules Category and should be unique.

Here's how you create a rule set:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: (select All Tasks from the Show menu) Manage Service Assignment Rules
2. From the Category drop-down list, select Generic Queuing Rules.

Note: Depending on what application you're using, Service Center, Help Desk, you select the corresponding generic category.

3. Click the **Add Row (+)** icon in the Rule Sets area.
4. Enter a name for the rule set.
5. In the Rules area, click the **Create (+)** icon, then do the following:
 - a. Enter a name and description in the respective fields.
 - b. (Optional) Set an effective start and end date.
 - c. Click the **Add Row (+)** icon in the Conditions area.
 - d. From the **Object** drop-down list, select **Any**.
 - e. From the **Attributed** drop-down list, select an attribute that you want to use in routing chats.
 - f. Select a value that must match to route your chats.

If you're setting up your chat requests to route to Oracle Digital Assistant, you'd select the Escalation Level attribute and set the value to equal 0 (zero). Then, you'd create a second rule within the same rule set to route to a live agent queue. The second rule would have the Escalation Level greater than 0 (zero). You can add other criteria to use for routing as needed.

6. When you're finished, click **Save and Publish**.
7. To confirm the rule was published, click the **Refresh** button.

A few things to note with routing rules

- Custom Attributes: If you've added a custom attribute and want to use it in routing rules, you must expose the custom field or object using the **Communication Channels > Manage Service Assignment Objects** page task on Functional Setup Manager.
 - o You can 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 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 didn't 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.

How do I set channel capacity?

Channel capacity indicates the maximum number of active interactions that an agent can handle. When the number of active interactions in an agent's queue reaches the specified capacity, the agent is considered to be 100 percent occupied.

For non-real-time interactions, you define what qualifying statuses are considered to be active when calculating an agent's capacity. Decide if any interaction in one of the specified qualifying statuses is considered to be an active interaction.

Here's some helpful pointers for setting up capacity for your agents:

- Oracle recommends not setting a high capacity number. The number should be relatively small so that routing can balance the work to agents.
- 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, let's suppose that the SR capacity is set to 30 and chat capacity is set to 2. And let's say an agent handles 15 open SRs and 2 chats. Then, that agent is considered to be 50 percent occupied for non-real-time work and 100 percent occupied for real-time work.
- Let's consider another example of how the workload is calculated when you have multiple work items such as SRs, activities, and escalations. Suppose an agent has a capacity of 5 for SRs and 2 for escalations. This doesn't mean the agent can work on 5 SRs and 2 escalations simultaneously. One SR takes up 20 percent of that agent's capacity, and one escalation takes up 50 percent of capacity.

To set channel capacity:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Communication Channels
 - Task: Manage Capacities

The Manage Capacities page displays capacities for real-time and non real-time channels. To change the total channel capacity, change the default values in the **Capacity** fields for the channels.

2. To provide queue owners and customer service managers the ability to change the channel capacities for individuals by overriding the global default capacities:

In the Override Individual Capacities region, select the **Enable Channel Capacities to Be Overridden for Individuals** checkbox.

3. To change the capacity of a non real-time work:
 - a. In the **Work Assignments** region, change 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 decide the capacity. For example, if the status is 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**.
4. To change the capacity of a real-time work, change the value in the **Capacity** field in the **Communication Channels** region.
5. Click **Save and Close**.

Note: When you export or import the functional setup data for the Service offering by using the export and import feature in Functional Setup Manager, the Manage Capacities setup is also exported or imported. But the value of the **Override Individual Capacity** checkbox isn't exported or imported. You must manually select or deselect the checkbox.

How do I configure the presence and availability privilege?

To set the presence and availability, an agent must be given the Manage Omnichannel Presence and Availability privilege.

This privilege is available by default to the following job roles:

To grant the **Manage Omnichannel Presence and Availability** privilege to any other role, 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.

For more information about copying job roles, see the *Oracle Fusion Cloud Customer Experience Securing Sales and Fusion Service* guide.

Note: After you add the privilege to the new role, you must associate a user to the new role.

Related Topics

- [Copy and Edit Duty Roles](#)

Prevent Work Starvation

Prevent Starvation of Low Priority Work Requests

Sometimes, a burst of high priority requests may prevent the low priority requests from being assigned to an agent for a long time.

Although high priority requests may come in later, they're assigned first, and the low priority requests are repeatedly kept waiting. That's because the agents are unable to keep up the pace to serve both.

As an administrator, you can dynamically raise the priority of low priority requests after a certain wait time. The benefits are as follows:

- The wait time for lower priority items is reduced, and this ensures a balance between high priority and low priority requests.
- Lower priority requests are served fairly, thereby improving satisfaction.

Configure Profile Options to Prevent Work Starvation

To prevent starvation of work requests having a lower priority, you must configure profile options.

The following table shows the profile options to configure for preventing work starvation.

Profile Options	Description
SVC_ENABLE_STARVATION_PREVENTION	Enable prevention of work starvation to avoid work requests having a lower priority from being ignored while assigning work items to an agent. The default value is No.
SVC_STARVATION_PREVENTION_NRT_INTERVAL	Specify the number of minutes after which the priority of an unassigned, non-real-time work request is increased for assigning to an agent. The default value is 180 minutes. Note: You must specify a value greater than the average handle time for a non-real-time work request in your company. For more information, see "Set a Profile Value Greater Than the Average Handle Time".
SVC_STARVATION_PREVENTION_NRT_STEP	Specify the step size to increase the priority of a non-real-time work request. The default value is 1.
SVC_STARVATION_PREVENTION_RT_INTERVAL	Specify the number of seconds after which the priority of an unassigned, real-time work request is increased for assigning to an agent. The default value is 30 seconds. Note: You must specify a value greater than the average handle time for a real-time work request in your company. For more information, see "Set a Profile Value Greater Than the Average Handle Time".
SVC_STARVATION_PREVENTION_RT_STEP	Specify the step size to increase the priority of a real-time work request. The default value is 1.

To configure the profile options for preventing starvation of work requests having a lower priority:

1. In the Setup and Maintenance work area, go to the following:
2. In the Manage Work Starvation Profile Options page, click **SVC_ENABLE_STARVATION_PREVENTION**.
3. In the SVC_ENABLE_STARVATION_PREVENTION: Profile Values region, set the **Profile Value** for the Site **Profile Level** as **Yes**.
4. Click **Save**.
5. Complete the following steps for the remaining four profile options in the table:
 - a. Select a profile option.
 - b. In the Profile Values region, select the Site **Profile Level**.
 - c. In the **Profile Value** field, retain the default value or modify the value according to your company's requirement.
 - d. Click **Save**.

Set a Profile Value Greater Than the Average Handle Time

The average handle time is the average duration of one interaction of an agent with .

You must set the profile for the SVC_STARVATION_PREVENTION_NRT_INTERVAL to a much higher value than the respective average handle times.

Related Topics

- [Configure Profile Options to Prevent Work Starvation](#)

FAQs About Service Requests and Queues

Frequently Asked Questions (FAQs) for Service Requests and Queues

This section contains some frequently asked questions (FAQs) when setting up and maintaining service requests and queues. Scroll or search to find answers to common questions.

Why aren't some service requests being assigned to the Default queue?

Let's say you're creating groovy scripts within Application Composer to perform validation checks, including user permission checks to determine who has access to update the fields in a service request (SR). Then you must ensure that you permit the internal user FUSION_APPS_CRM_ESS_APPID to update SRs.

The FUSION_APPS_CRM_ESS_APPID internal user performs operations on behalf of Omnichannel. If the internal user isn't given permission to update SRs from within the groovy script, then Omnichannel is unable to set the Default queue as the target queue. As a result, SRs aren't assigned to the Default queue.

Related Topics

- [How do I create queues?](#)
- [Overview of Groovy Scripts](#)

What happens if a newly created request isn't assigned to a queue?

If a is created without an associated queue, and Omnichannel is enabled, then the routing feature selects a queue.

If the queue is automatic, then the routing feature assigns the associated agent for the . If is created with an associated queue, then the routing feature selects only the agent.

What happens if an associated agent is removed from an open request?

If an associated agent is removed from an open that's in a status of New or any other active statuses specified in the Manage Capacities page, then the following happens:

- The routing feature selects a new agent for the from the resource pool of that queue, based on the capacity of the associated agent.

Note: A new agent is selected only if Omnichannel is enabled and the queue is automatic.

Related Topics

- [How do I set channel capacity?](#)

What happens if an open request is disassociated from the current queue?

If an open is disassociated from the current queue, then the routing feature assigns the to a new queue, provided the following conditions are true:

- Omnichannel is enabled.
- The **Assigned To** value of the is empty.

The is subsequently assigned to an associated agent if the queue is automatic and any assignment rules are rerun. If the disassociated is in a New status or any other active status specified in the Manage Capacities page, then the agent capacity is calculated before assigning the to an agent.

Note: If you disassociate from the current queue, then you must clear the **Assigned To** value to ensure that rerouting takes place.

What happens if a request is reopened?

Here's what happens when a is reopened:

- If the reopened is already associated with an agent, then the is reassigned to the same agent.
- If the reopened isn't assigned to any agent, and Omnichannel is enabled, then the is assigned to a queue based on the associated assignment rules.

If the queue is automatic, then the is assigned to an associated agent. are assigned based on the set channel capacity for an agent. Only if an is in the New, In Progress, or other statuses specified in the **Qualifying Status Values** column in the Manage Capacities page in Setup and Maintenance, it counts for capacity. If the isn't in one of these statuses, it doesn't count for capacity, and the isn't assigned to the agent.

Note:

- If the status of changes, then it counts for capacity for the agent to whom it's assigned. For example, when the status changes from Waiting for to In Progress.
- If the isn't assigned to an agent, the routing service starts looking for an agent.

Related Topics

- [How do I set channel capacity?](#)

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.

10 Service Entitlements

Overview of Service Entitlements

What are entitlements?

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.

Here's the ready-to-use milestones you get with the application:

- **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, you can also 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. You can apply milestones to a service request by defining the coverages as follows:

- **Standard Coverages**, which act as templates.
- **Default Coverages**, which associate the standard coverage templates with a customer (customer-level default), or specify a standard coverage template as a global default that's applicable to all service requests.
- **Subscription Coverages (or Service Contracts)**, which you can optionally create to apply coverages only to specific assets or products owned by a customer.

Note: New customers are automatically opted-in to use Subscription Management instead of Service Contracts. For existing customers implementing service entitlements for the first time, Oracle recommends the use of the subscription coverages. Customers already using Service Contracts are still supported, however new features such as product-based service level agreements (SLAs) are only available with Subscriptions. For more information, see [How do I enable subscription coverages for entitlements?](#). When the same milestone exists in coverages at multiple levels, the subscription (or contract) values take priority over customer-level default values, which in turn take priority over global default values.

To apply milestones for specific assets or products, you must set up Subscriptions (or Service Contracts), opt-in to use installed base assets from the common asset model, and allow an installed base asset to be picked on a service request. If you use Subscriptions (or Service Contracts), you specify the standard coverage on a contract line and identify which installed base assets or products are covered. When an installed base asset or product is selected on a service request, contract-based coverages are applied to the milestone determination. When a service request specifies an installed base asset or product that's covered on a subscription (or service contract), that corresponding coverage is automatically considered when applying milestones to the service request.

Note: For any service requests without a covered asset or product, the existing global and customer default coverages apply.

Overview of the 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.

The **Milestone** object includes a **Status** field that indicates whether the milestone is being actively tracked, is paused, complete, or has been canceled. The **Milestone** object also includes two flags that are commonly used to trigger object workflow according to your business needs, for example to send an email notification to proactively notify the agent, or escalate the service request:

- The **WarnedFlag** is updated to **Y** when the milestone's time to expiration is within the warning threshold specified in the standard coverage.
- The **ComplianceFlag** is updated to **N** when the milestone reaches the due date or the time before completion, representing a milestone that has expired (it's no longer compliant).

In case you're wondering where you can get more details about the setup and configuration of milestones and coverages:

- The topic "Set Up Milestones and Coverages" provides the details to set up milestones and coverages using the delivered milestones and coverage criteria.
- The topic "Configure Milestones and Coverages" describes how to expand milestones and coverages to adapt them to your business needs.

Milestones and Coverages

How do I set up milestones and coverages?

You must set up milestones and coverages so that the ready-to-use milestones are automatically applied to service requests and monitored.

New customers are automatically opted-in to use Subscription Management instead of Service Contracts. For existing customers implementing service entitlements for the first time, Oracle recommends the use of the subscription coverages. Customers already using Service Contracts are still supported, however new features such as product-based service level agreements (SLAs) are only available with Subscriptions. For more information, see [How do I enable subscription coverages for entitlements?](#)

Here's the steps you require for this set up:

1. Manage milestone configuration (optional)
2. Create a coverage schedule (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)

Overview of Milestones and Coverages Configuration

The First Response Metric and Resolution Metric milestones are preconfigured for every implementation.

These 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.

After you set up the milestones and coverages using the standard process steps described in the "How You Set Up Milestones and Coverages" topic, you can cover a wider variety of scenarios and adapt milestone functionality to your specific business needs. Here's where you can get detailed information about the advanced configuration for milestones and coverages:

- The "Set Up Additional Condition Columns for Standard Coverage Entitlement Rules" and "Set Up Administrator-defined Milestones" sections provide information about managing milestone conditions, as well as creating your own milestones.
- The "Configure Milestones for Multiple Business Units" section provides information about how to configure milestones for multiple business units.
- The "Troubleshoot with Milestone Diagnostics" section provides information about reviewing milestone diagnostics for troubleshooting.

What's a Coverage Schedule and how does it work?

A coverage schedule is a service calendar containing detailed time intervals that identify when a request is expected to be worked.

Coverage schedules provide flexibility to define business hours and specific intervals during the year when different operating hours are offered. For example, you can define a specific interval such as a particular week or seasonal time period where you might extend service hours on specific days. You can also treat holidays or exceptions more specifically, such as offering shortened hours on some holidays, instead of a full day inclusion or exclusion.

To identify the due date and time for a milestone, time outside the specified intervals is skipped. For example, if your coverage schedule 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 schedules can also contain holidays, nonworking times, or extended working hours, which identify days to be skipped or hours to be added when deciding the due date. You must configure the coverage schedule exceptions before you create or change a coverage schedule. In the previous example, when a milestone started 4:00 pm Friday and had 120 minutes until due, if Monday was identified as a holiday, the milestone would be due at 10:00 am on Tuesday instead.

Here's how the application calculates the **Due Date** and **Time Remaining** fields that are displayed on the Service Request Milestone Details page:

- **Due Date:** The due date for a milestone is calculated using the schedule that's associated with the applicable entitlement or coverage rule. The schedule defines whether there are any hours of the day or week when the clock isn't running for the organization, such as evenings, holidays, lunch breaks, and so on.
When a service request is updated, the application looks at the coverage rules and the associated schedule, calculates the date and time the milestone expires, and assigns the milestone to the service request. The milestone due date is calculated based on the time the milestone starts, which might not be the same as the

time the service request was created. The application displays this due date for that milestone on the Service Request Milestone Details page.

If the SR attributes change so that the milestone goes into a paused state, the clock stops running. The application captures the time spent on the milestone up until the milestone was paused, for future use. The due date for the milestone is no longer displayed, because it can't be determined until the milestone resumes the countdown. After the pause is lifted and the countdown resumes, the application recalculates the milestone due date. The recalculation is based on the original time you specified for that milestone when it was created, minus the cumulative time spent by the agent up to that point.

- **Time Remaining:** The **Time Remaining** field displays its value depending on whether the milestone is actively counting, paused, or overdue. The application doesn't show time remaining for milestones that are complete or canceled.

When the milestone is actively counting down, the exact due date is known. The **Time Remaining** field shows an approximation of the amount of time between the due date and the current time. This approximation is expressed in the largest indivisible unit of time and rounded down. For example, if the current time is 3:00 PM and the milestone is due at 4:05 PM, the time remaining shows 1 hour. If the current time is 10:00 AM and the milestone is due the next day at 11:00 PM, the time remaining shows 1 day.

When the milestone is paused and the due date can't be determined, the **Time Remaining** field identifies only the time left in the milestone. The time left is determined by calculating the total time specified in the coverage, minus the time spent so far. For example, a milestone that was set for 24 hours (based on the coverage rules), and that was active for 10.5 hours before it paused, will have 13.5 hours remaining for the duration of the time it's paused. The **Time Remaining** field uses the same approximation method as described for active milestones, so 13.5 hours is rounded down to show 13 hours. Again, the actual due date isn't in 13 hours, because it's unknown when the counting will resume and therefore how the schedule might affect the due date. The 13 hours is given as an approximation of the working time you'll have once the milestone restarts.

When the milestone is overdue but still not complete, the Time Remaining field shows how much time has elapsed since the milestone expired. The profile option `ORA_SVC_SLA_INCLUDE_PAUSED_MINS_IN_TIMEREMAINING` indicates whether this calculation considers paused minutes in the calculation.

Excluding paused minutes ignores whether the milestone has paused after expiration, and results in a more exact calculation of actual time elapsed and improved performance. Including paused minutes considers how much time was spent in the paused state after expiration and reduces the overdue time remaining value by that amount. This requires more calculations, is less consistent with the calculation before expiration, and can impact performance.

Note: Oracle recommends setting this profile option to N and starting with release 23D, all implementations will have the value set to N. You can then change the value to Y if you want for backward compatibility.

A schedule named **24 by 7** is preconfigured for all implementations. This schedule doesn't specify holidays or downtime, so the due date for a milestone that uses this schedule, is calculated without skipping any time.

Related Topics

- [How do I set up milestones and coverages?](#)
- [How do I create a coverage schedule?](#)
- [How do I create default coverages?](#)
- [How do I create standard coverages?](#)

How do I enable subscription coverages for entitlements?

Oracle recommends using subscription coverages. New customers are automatically opted-in to Subscription Management instead of Service Contracts. Existing customers who are implementing service entitlements should consider subscription coverages.

The **Manage Service Entitlements Using Subscription Coverages** feature is enabled by default for new instances. If this feature is enabled, the application uses Oracle Subscription Management Cloud to define entitlement calendars, coverages, and associated setup, and uses this setup at runtime to identify service request milestones.

Here's how you can enable this feature:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
2. Click **Change Feature Opt In**.
Select the **Enable** checkbox in the Service Entitlements row.
3. Click **Done**.

How do I create a coverage schedule?

To create your own coverage schedule calendars with the intervals and holidays specific to your business, do the following:

1. Sign in to the application as a sales administrator.
2. On the **Navigator**, click **Subscription Management**.
The Subscriptions page is displayed.
3. Click **Subscription Configuration**.
The Subscription Configuration page is displayed.
4. To create a coverage schedule, click **Entitlements Management > Manage Availability**.
The Manage Availability page is displayed.

Note: If you're using Service Contracts, then you navigate to this page by clicking **Navigator > Contract Management > Contracts > Tasks > Manage Availability**.

5. (Optional) To add the holidays, nonworking times, or extended working hours that are applicable to the coverage schedule, click the **Exceptions** tab.
 - a. Click **Create** in the **Exceptions** pane to create a new coverage schedule exception, or select an existing entry and click **Duplicate** to copy and update it.
 - b. In the **Availability Exceptions** section, click **Add Event**.
 - c. Enter the name of the exception event, the date and times of the event, and the availability during the event.
 - d. Click **Save**.
 - e. (Optional) To delete an exception, click **Delete** in the **Exceptions** pane.

6. On the **Schedules** tab, click **Create** in the **Schedules** pane to create a new coverage schedule, or select an existing entry and click **Duplicate** to copy and update it.
7. Specify the following details:
 - **Schedule Name:** Name of the coverage schedule.
 - **Time Zone:** Time zone for which you're creating the coverage schedule.
8. In the **Date Intervals** section, create a coverage schedule interval with details of when the coverage schedule starts, ends, and the days of the week that are part of the interval.
 - **Start Date** and **End Date:** Start and end dates for the coverage schedule interval.
 - **Availability:** Working days, hours, and times, and break times for the selected days.
9. (Optional) To include exceptions in the coverage schedule, select an exception from the **Exceptions** drop-down list.
10. (Optional) To delete a schedule, click **Delete** in the **Schedules** pane.
11. Click **Save**.

You can create multiple coverage schedules 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** schedule, but a lower severity service request milestone might use a different schedule that's only worked during normal business hours. For more details about using coverage schedules in standard coverages, see the "Create Standard Coverages" topic.

How do I create standard coverages?

The standard coverage acts as a template that contains entitlement rules, which specify when milestones are due and for what conditions.

For example, high severity service requests may have earlier milestone times, and use a schedule with longer work hours and fewer holidays, than lower severity service requests.

1. On the Subscription Configuration page, click **Entitlements Management > Standard Coverage**.

The Manage Standard Coverage page is displayed.

2. Create a new coverage and select **Subscription Entitlements** as the **Entitlement Type**.

The Edit Standard Coverage page is displayed.

Note: If you're using Service Contracts, then you navigate to this page by clicking **Navigator > Contract Management > Contracts > Tasks > Standard Coverage**. Then select **Contracts Service Entitlements** as the **Entitlement Type**.

3. Select **Actions** > **Add Entitlements** to configure the entitlement rules structure. The **Entitlement Rules** dialog box is displayed.

Service request severity is automatically included as a condition column. If you want to vary milestones based on the service request channel, optionally, select **Channel Type** as an optional condition column.

- a. 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 shouldn't have a first response milestone tracked, you can deselect **First Response Metric** and **First Response Warning Threshold** from **Optional Results Columns**.

- b. Click **Finish** to apply the entitlement options and complete the configuration of the entitlement rules. The Edit Standard Coverage page is displayed.
4. 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.

The algorithm applying milestones to service requests requires that coverages never have more than one row with condition columns that match the service request. If you leave a condition column value blank, the null value acts as a wildcard, and then is matched against any service request value. While there are valid scenarios where this can be used, it's important to avoid having two lines that could match the same service request.

The following table shows an example of a valid use of a null value in standard coverage condition columns:

- In this valid example, a service request with high severity will only match line 1, regardless of the channel.
- A service request with low severity will only match line 5, regardless of the channel.
- The medium severity service request can only match one of lines 2 through 4, depending on the channel type.

Line Number	Severity	Channel Type
1	High	Blank
2	Medium	Phone
3	Medium	Email
4	Medium	Chat
5	Low	Blank

Line Number	Severity	Channel Type

The following table that shows an example of an invalid use of a null value in standard coverage condition columns:

- This invalid example shows a situation where the medium severity service request with a phone channel matches lines 2 and 4.
- Similarly, a medium severity service request with the email channel matches lines 3 and 4.
- In both cases, the service request won't have any milestones applied because more than one line from a single coverage matches the service request.

Line Number	Severity	Channel Type	Comments
1	High	Blank	
2	Medium	Phone	Medium, Phone would match this line
3	Medium	Email	Medium, Email would match this line
4	Medium	Blank	Medium, Phone would also match this line Medium, Email would also match this line
5	Low	Blank	

To configure the behavior of null values in a coverage, select the **Allow Null** and **Null is Wildcard** check boxes for the selected matrix class. You will find these check boxes on the Edit Matrix Class: Subscription Entitlements page, **Condition Columns** section. For more information about using these properties on condition columns, see the section "Set Up Additional Condition Columns for Standard Coverage Entitlement Rules".

- 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 schedule to use for calculating the milestones' due date.
 - ii. For each milestone, enter the metric in minutes, which is used with the schedule 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's **WarnedFlag** is set to **Y**.
- 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 schedule) are used to calculate the milestone due date.

How do I create default coverages?

You use default coverage to apply a standard coverage template to specific customers, or globally to 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.

1. On the Subscription Configuration page, click **Entitlements Management > Default Coverage**.

The Manage Default Coverage page is displayed.

Note: If you're using Service Contracts, then you navigate to this page by clicking **Navigator > Contract Management > Contracts > Tasks > Default Coverage**.

2. Click **Create** to add a new coverage.
3. 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.

4. Enter the start and end date for the coverage.
5. Click **Save**.

How do I monitor milestones?

You configure the Monitor Service Request Milestones scheduled process to monitor the service request milestone status, and set the job frequency to run the job after the specified time interval.

This 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.

How do I expose milestones in the SR 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. Note: The time remaining field is an approximation and it's rounded down so that agents don't think they have more time than they actually do. To expose current milestone fields on the service request layout, create a user-defined layout in Application Composer, and add the MilestoneID field to the Service Request Summary page.

How do I enable email notifications to monitor milestones?

You can configure email notifications to warn required teams and managers that a milestone is about to expire, or email notifications can be used to escalate a service request when a milestone expires.

After you set up the scheduled process, you can configure object workflows to perform actions when a milestone's WarnedFlag is set to **Y** or ComplianceFlag is set to **N** (when the milestone expires).

1. Sign in to the application as an administrator.
2. Navigate to **Application Composer** and then click **Common Setup > Email Templates**. 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 **Navigator > Configuration > 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. Specify the **Event Point** and **Condition** for the milestone object workflow.
For example, to trigger a workflow action when the `WarnedFlag` is set to **Y**:
 - a. For the **Event Point** option, select **When a record is updated**.
 - b. In the **Condition** field, enter the following expression:

```
if (isAttributeChanged('WarnedFlag') && WarnedFlag=='Y')
return true;
```
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, such as looking up the assignee and assignee manager details for sending the notification. Here's an example script you can use to get the assignee and assignee manager details and populate the **Address** field for the email notification.

```
def resourceVO = newView('Resource')
resourceVO.appendViewCriteria("PartyId = ${AssigneePartyId}")//party id of SR assignee
resourceVO.executeQuery()
```

```
def mgrPartyId
while (resourceVO.hasNext()) {
    def curResourceRow = resourceVO.next()
    mgrPartyId = curResourceRow.ManagerPartyId
}
return mgrPartyId
if (mgrPartyId != null) {
    def resourceMgrVO = newView('Resource')
    resourceMgrVO.appendViewCriteria("PartyId = ${mgrPartyId}")
    resourceMgrVO.executeQuery()
    def mgrEmailId
    while (resourceMgrVO.hasNext()) {
        def curResourceMgrRow = resourceMgrVO.next()
        mgrEmailId = curResourceMgrRow.EmailAddress
    }
    return mgrEmailId
}
```

Can I view an expired milestone sample script?

This is sample code to trigger a notification to be sent to the recipient specified in the Notification Preferences page when a milestone expires.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Service Request > Milestone > Server Scripts.

Note: This example uses `WarnedFlag`. However, you can also use `complianceFlag` for similar results.

```
/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Update in Database
 * OBJECT: Milestone (child of Service Request)
 * Use Case: Send a notification to the assignee when an SR milestone is expiring.
 * Note: You can override the Notification Text and Recipients using the Notification Preferences page.
 */

if (isAttributeChanged('WarnedFlag') && WarnedFlag=='Y') {
    try {
        def recipientPartyId = SrAssigneePartyId
        def messageText = 'An SR milestone notification (default message).'

        if (recipientPartyId) {
            //Call to send notification
            adf.util.sendNotification(adf, messageText, recipientPartyId)
            //Log a confirmation that the notification has been sent. Logs can be viewed in 'Runtime Messages'.
            //println("Notification sent to " + recipientPartyId + " because the SR was escalated.")
        } else {
            println("No Assignee associated with this SR")
        }
    } catch (e) {
        //Log the failure in groovy logging. Logs can be viewed in 'Runtime Messages'.
        println("Failure to trigger notification from Groovy Script " + e.getMessage());

        // Throwing validation exception will show the message on the UI. This is not recommended for published
        // sandboxes.
        // The following code is one of many to illustrate identifying an error in trigger from the UI.
        // Replace <triggerName> with the trigger name you specified when creating this trigger.
        // throw new oracle.jbo.ValidationException('Failure to trigger <triggerName> Notification from Groovy
        // Script: ' + e.getMessage())
    }
}
```

}

Set Up Additional Condition Columns for Standard Coverage Entitlement Rules

How do I set up more condition columns for standard coverage entitlement rules?

The application has a default entitlement type called Subscription Entitlements (or Contracts Service Entitlements), which is a matrix class, and a corresponding service mapping that you select when you create your standard coverage.

This default 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 request to determine which line is applicable, and the result columns display the schedule, milestones, and thresholds that are applied.

You can use other columns from the 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.

How do I create a service mapping?

Here's how you create a service mapping:

1. On the Subscription Configuration page, click **Dynamic Matrix Configuration > Manage Service Mappings**. The Manage Service Mappings page is displayed.
2. Click **Subscription Entitlements**. The Edit Service Mapping: Subscription Entitlements page is displayed.
Note: If you're using Service Contracts, then you navigate to this page by clicking **Navigator > Contract Management > Contracts > Tasks > Manage Service Mappings > Contracts Service Entitlements**.
3. On the **Entities** tab, select .
4. In the **ServiceRequest: 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**.
 - Ensure that **Write** is disabled for the selected attribute.
5. Click **Save**.

6. Click the **Sources** tab (on the same page, Edit Service Mapping: Subscription Entitlements).
7. In **Source**, select .
8. Select **ServiceRequest** in the tab.
9. On the **Entitlement: Details > Attribute Mappings** tab, click **Add Row** to create the mapping for the new attribute.
10. Select the attribute that you created in step 4 from the **Attribute** drop-down list and enter the related **View Object Attribute** of the .
Note: Ensure that **Write** is disabled for the selected attribute.
11. Click **Save**.
12. Click the **Services** tab (on the same page, Edit Service Mapping: Subscription Entitlements).
13. Select **ServiceRequest** in the **GetSubscriptionEntitlements: Details > Entities** tab.
14. In the **ServiceRequest: Entities** section, click **Add Row** to create the mapping for the new attribute.
15. Select the attribute that you added in Step 4 from the **Attribute** drop-down list and enter the values.
Note: Ensure that **Write** is disabled for the selected attribute.
16. Click **Save and Close**.

How do I set up a milestone based on a field I created?

The default entitlement type Subscription Entitlements (or Contracts Service Entitlements), is a matrix class. A corresponding mapping is selected when you create your standard coverage. Here's how you create a matrix class and add more conditional columns.

This default 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 schedule, 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.

1. On the Subscription Configuration page, click **Dynamic Matrix Configuration > Manage Matrix Classes**. The Manage Matrix Classes page is displayed.
Note: If you're using Service Contracts, then you navigate to this page by clicking **Navigator > Contract Management > Contracts > Tasks > Manage Matrix Classes**.
2. Duplicate the **Subscription Entitlements** (or **Contracts Service Entitlements**) matrix class. You can also edit a matrix class you created before, or create a new matrix class.
3. 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.
4. Enter the **Name** of the column and the **Source Code Name**.
5. In the **Compare to Attribute** column, select the attribute that you created in the previous service mapping procedure.
6. In the **Domain** column, click **Edit Domain** and select a domain value corresponding to the attribute you're mapping.

7. To allow null values to be specified in the coverage for this column, select the **Allow Null** checkbox, and optionally, select the **Null is Wildcard** checkbox.

Then select the appropriate value for the **Null is Wildcard** option, based on the intended behavior described in the following table. The **Severity** condition column is used as an example in the following table.

Allow Null Selected?	Null is Wildcard Selected?	Add a Coverage Row with Blank Value?	Coverage Rows Applied on SR with Blank Value Specified?	Coverage Rows Applied on SR with Non Blank Value Specified?
N	N	Not possible	No rows applied	When explicit match between SR and coverage row.
Y	Y	Y	Blank row applied	- If the SR severity isn't defined in a row, the blank coverage row is applied. - If the SR severity is defined in a coverage row, an error occurs and no values are applied. Both the blank row and the matching row are considered a match, but coverages must be set up so that only one row from a specific coverage can ever be matched. If two rows match, it results in an error condition and no rows being applied. For more information, see the step 5b in the topic "Create Standard Coverages".
Y	Y	N	No rows applied	When explicit match between SR and coverage row.
Y	N	Y	Blank row applied	When explicit match between SR and coverage row.
Y	N	N	No rows applied	When explicit match between SR and coverage row.

Allow Null Selected?	Null is Wildcard Selected?	Add a Coverage Row with Blank Value?	Coverage Rows Applied on SR with Blank Value Specified?	Coverage Rows Applied on SR with Non Blank Value Specified?

8. Click **Save and Close**.

The matrix class displays in the **Entitlement Type** drop-down list when you create a standard coverage.

Set Up Administrator-Defined Milestones

Overview of Administrator-Defined Milestones

In addition to using the predefined milestones for First Response and Resolution, you can also define additional milestones to track requests. This lets you configure the application effectively, based on your own business processes and service standards.

The section "Set Up Additional Condition Columns for Standard Coverage Entitlement Rules" described how you can use other columns from the request as condition columns for 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**.

Add the Result Columns to the Matrix Class

Here's how you can add the result columns to the matrix class:

1. On the Subscription Configuration page, click **Dynamic Matrix Configuration > Manage Matrix Classes**. The Manage Matrix Classes page is displayed.

Note: If you're using Service Contracts, then you navigate to this page by clicking **Navigator > Contract Management > Contracts > Tasks > Manage Matrix Classes**.

2. Duplicate the (or **Contracts Service Entitlements**) matrix class. You can also edit a matrix class you previously created, or create a new matrix class.
3. 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.
4. Enter names for the administrator-defined metric and its corresponding warning threshold.

Code values are automatically populated in the **Source Code Name** field. The names and codes must be unique for each milestone and its corresponding threshold value (**Milestone Code** and **Threshold Code** must be unique and should not be used already in existing milestone configurations). You should copy and save these values, because you need **Source Code Name** in the Manage Algorithms page and **Name** in the Manage Service Milestone Configuration task page.

5. In the **Domain** field, retain the default value **Number** because it denotes the time in minutes.
6. Click **Save and Close**.

Modify the Algorithm to Include the New Milestone and Threshold Columns

Here's how you can modify the algorithm to include the new milestone and threshold columns you created:

1. On the Subscription Configuration page, click **Dynamic Matrix Configuration > Manage Algorithms**.

The Manage Algorithms page is displayed.

Note: If you're using Service Contracts, then you navigate to this page by clicking **Navigator > Contract Management > Contracts > Tasks > Manage Algorithms**.

2. Select the **Get Subscription Entitlements** (or **Contracts Get Service Entitlements**) row, with the algorithm version **1** and status **Published**.
3. Select **Actions > Create Version**. A new version of the algorithm is created, with the status **In Progress**.
4. Select the new version of the algorithm. The Edit Algorithm page is displayed.
5. On the **Algorithm** tab, select **Look up matching Entitlement Rule**.
6. In the **First Row Actions** section, click in the **Actions** field.
7. Copy the existing algorithm and append it. For example:

```
if(Matrix.hasProperty("NextResponseMetricCode")) {  
rule= EntitlementResults.insert([EntitlementResultId: getNextId()])  
rule.EntitlementId=Entitlement.EntitlementId  
rule.ScheduleId=Matrix.CalendarCode  
rule.ResultName= 'NextResponseMetricCode'  
rule.ResultValue= Matrix.NextResponseMetricCode  
rule.ResultThresholdValue=Matrix.NextResponseWarningMetricCode  
}
```

Note: If you're using Service Contracts, use `rule.CalendarId=Matrix.CalendarCode` instead of `rule.ScheduleId=Matrix.CalendarCode`.

8. 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.

9. Click **Save and Close**.
10. Navigate to the Manage Algorithms page.
11. Select the modified version of the algorithm, and click **Actions > Publish**. The **Status** of the algorithm is updated to **Published**.

How do I add the milestone to the Manage Service Milestone Configuration Task?

Here's how you can add the milestone to the Manage Service Milestone Configuration task:

1. In the Setup and Maintenance work area, go to the following:

The Manage Milestone Configuration page is displayed.

2. Click **Create Milestone** and specify the following milestone details:
 - **Milestone Label:** Enter the name of the milestone that should be displayed on the request.
 - **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 can't be modified.
 - **Milestone Type:** Select from the following types, which determine whether milestones are reopened or repeated once complete.
 - **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 can't 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 update is posted to the request. Once a response is sent, the milestone is complete, but upon receiving a new update, another **Next Response Due** milestone is created and tracked.
 - **Can't be repeated:** A milestone of this type can't be reopened or repeated once it's marked complete. 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's applied to the milestone. The names are defined in the matrix classes and can't be modified.
 - **Business Unit Name:** Leave as the default business unit (BU) unless you're configuring milestones for multiple business units. For more information, see the "Configure Milestones for Multiple Business Units" in this guide.
 - **Allow Due Date Override:** Select this check box to let agents modify a milestone's due date on the Service Request Milestone Details page (in the **Due Date** field).
- Note:** If a customer wants to audit agent updates to the due date, the administrator must enable audit on the **Overridden Due Date** attribute.
- **Allow Agent Completion:** Select this check box to let agents manually acknowledge that they have completed a milestone and close it. On the Service Request Milestone Details page, if an agent selects the **Agent Completed** check box and saves the milestone, the milestone status is updated to **Complete**. Milestones that can be reopened are reopened if the start condition is met again.

The administrator can specify which milestones allow due date updates and which allow the update to specify a milestone has been completed. These attributes are independent of each other, so that some milestones can allow only moving the due date, others might allow the agent to mark it complete but not move the due date, and still others may allow both types of updates.

3. 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.
 - o Each tab has a text box that enables you to provide a common description that's displayed in the request details when the milestone is applied to the request. This description enables an agent to understand when a milestone starts counting, when it pauses, and when it's considered complete.
 - o 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. The **Attribute** column lists administrator-defined fields and a subset of the service request fields that are available to use as conditions. Alternately, you can specify a static condition that doesn't require an operator or value. You can also use object functions written in Groovy, to express more complex start, pause, and stop criteria for milestones.
 - 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.
 - You can specify a logical condition in the form `<attribute> <comparison> <operator values>`.

For example, you can specify `status Type (attribute) Is One Of (operator) closed, Resolved (values)` as the completion criteria for Resolution Milestone. You can also specify a static predefined condition such as `User Sends Response`, which evaluates to true as soon as any user sends a response message.

When an object function is written on the request object in Application Composer to return a Boolean value, the function is available as start, pause, and completion criteria. Similar to the static conditions, when an administrator-defined object function is selected, the operator and values aren't required. When the function returns true, the start, pause, or completion criteria specifying that function is triggered.

- o 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.

Configure Milestones for Multiple Business Units

What are the milestones for multiple BUs?

You can configure milestones using the Manage Service Milestone Configuration or Manage Service Milestone Configuration for Business Unit task.

As an administrator, you can create and manage milestones for service requests using the **Manage Service Milestone Configuration** task in **Setup and Maintenance**. However, when the **Business Units** feature of **Service Offering** is enabled, milestones must be defined for each business unit (BU) that intends to track them.

Here's how milestones are applied when BUs are enabled:

- If a service request is assigned to a BU, and milestones are defined for that BU, then the milestones are applied.
- If a service request is assigned to a BU, but no milestones are defined for that BU, then the milestones aren't applied.

- If a service request isn't associated with any BU, and a default BU is defined for that organization, then the milestones defined for the default BU (if any) are applied.

How You Configure Milestones for Multiple BUs

In a multiple BU environment, you must configure milestones for each BU that intends to track them.

This gives each BU the flexibility to track a different set of milestones, or to track the same milestone using different start, pause, and stop criteria, depending on the business requirements. You can configure milestones for each BU in the following ways:

- To apply an existing milestone to a new BU, or to create a new milestone for a BU, use the **Manage Service Milestone Configuration** task in **Setup and Maintenance**.

Using this task, you create a new milestone configuration, and specify the existing or new milestone code with the new BU. You can specify the same or different start, pause, and stop criteria, depending on the requirements of the BU.

- To define multiple milestones for a specific BU, use the **Manage Service Milestone Configuration for Business Unit** task in **Setup and Maintenance**.

When using this task, you must specify the **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.

Note: Before configuring milestones for the first time in a newly-provisioned multiple BU environment, the administrator must associate the default BU with the ready-to-use milestones, as described in the following topic.

Configure Milestones for the First Time in a Newly-Provisioned Environment

When you're first configuring milestones in your newly provisioned, multiple BU environment, after the default BU has been set up for your organization, you must associate the default BU with the ready-to-use milestones.

This required association is a one-time task to enable milestones to be configured for multiple BUs.

1. In the Setup and Maintenance work area, select the **Manage Service Milestone Configuration** or **Manage Service Milestone Configuration for Business Unit** task.

The Select Default Business Unit page is displayed.

2. From the **Business Unit Name** drop-down list, select the default business unit to assign to the ready-to-use milestones, and click **Apply**.

Milestone configurations corresponding to the selected business unit are displayed.

Troubleshoot with Milestone Diagnostics

Overview of Milestone Diagnostics

Administrators can use the Request Milestone Diagnostics page to validate or troubleshoot your milestone rules. The administrator can run a milestone diagnostics report for a specific request and view the full evaluation of milestone rules.

The milestone diagnostics report provides the following information:

- Which milestones were applied?
- Which milestones and coverages were considered but not applied?
- From which coverage did an applicable milestone originate?
- Did the milestone apply or not apply based on the associated business unit?

How do I view the milestone diagnostics report?

Here's how you can view the milestone diagnostics report:

1. In the Setup and Maintenance work area, go to the following:
2. On the Manage Milestone Configuration page, click **Actions > View Milestone Diagnostics**.

The Service Request Milestone Diagnostics page is displayed.

3. Enter the request number in the **Service Request** field.
4. Click **Check Milestones**.

The milestones that are applicable to the specified request are displayed in a table, with the corresponding metrics and the following details:

- **Defined for BU:** Indicates whether the milestone configuration definition was created for the applicable business unit.
- **Applicable to SR:** Indicates whether the milestone from the specified coverage would be applied to the request.
- **Notes:** Provides more information when the milestone isn't applicable to the request.

11 Set Up Service Business Units

How are business units for set up for Service and Help Desk?

With business units (BUs), you can deploy more than one service center within a single instance of your service or help desk application. This topic gives an overview of the steps that you must perform to complete the business units setup.

You can use multiple BUs in the following ways:

- Segment SRs between BUs so that users can search and identify SRs from multiple BUs.
- Use product catalogs, categories, channels, and email templates specific to a BU.
- Assign SRs to queues by writing rules based on BU.
- Create service request BI reports specific to a BU.

Currently, the following objects aren't supported by multiple BUs in Service: accounts and contacts, users, resources, and lookups.

To set up business units in Service, you must perform the following tasks in the given order.

Step	Description
Setting Up Business Units for Service	Add the Manage Business Unit functional area to the Service offering by using the Change Feature Opt In link in Setup and Maintenance.
Manage Common Profile Options	Set the profile options to enable the multiple-BU functionality in the Manage Common CRM Business Unit Profile Options task.
Manage Internal Resource Organizations	Define internal resource organizations to be associated with the BU.
Manage Resource Organization Hierarchies	Add the internal resource organizations to the internal resource organization hierarchy.
Create Business Unit	Create a BU to be associated with the resource organization in the Manage Business Unit Task.
Associate Resource Organization to Business Unit	Associate the internal resource organization to the BU you created. Use the Resource Directory.
Create Employees	Add users to the organization in the Users, Roles and Delegations task, and make one of the resources a manager for the organization.
Reset Passwords for Users	Reset the password for the users.

Step	Description
Set the Scope in Service Setup Tasks and complete Service Setup tasks.	Set the scope for Service tasks and set up the remainder of the Service offering.
Add additional BU fields in the SR if users are associated with multiple BUs.	Change the layouts of the SR pages if required. Use Application Composer to include multiple BU fields in the SRs. Note: Only required if a user is associated with multiple business units.
Configure Multiple Business Units with Digital Customer Service	If you implement multiple business units for Digital Customer Service, you must complete some additional configuration.

How do I enable multiple Business Units Fusion Service?

To enable multiple business units (BUs) for Service , you must set the following profile options:

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 business unit for your service application.

Note: You must ensure that the value of the HZ_DEFAULT_BU_CRM profile option doesn't remain blank, and the value is set to a BU. Otherwise, agents might see an error while creating SRs.

To set the profile options for Service BUs:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Company Profile
 - o Task: Manage Common CRM Business Unit Profile Options

The Manage Common CRM Business Unit Profile Options page shows the two profile options.

2. Click the **HZ_ENABLE_MULTIPLE_BU_CRM** profile option.
3. In the HZ_ENABLE_MULTIPLE_BU_CRM: Profile Values region, set the **Profile Value** for the Site **Profile Level** to **Yes**.
4. Click **Save and Close**.
5. Click the **HZ_DEFAULT_BU_CRM** profile option.
6. In the HZ_DEFAULT_BU_CRM: Profile Values region, specify the **Profile Value** for the Site **Profile Level**.
7. Click **Save and Close**.

Manage Internal Resource Organizations for Service BUs

For every business unit (BU) that you set up, you can define internal resource organizations to be associated with the BU.

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 for a Service BU:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Users and Security
 - o Task: Manage Internal Resource Organizations
2. Click **Create** to add a new resource organization.
3. Select the **Option 2: Create New Organization** option to create a new organization.
4. Click **Next**.
5. On the Create Organization: Enter Basic Information page, enter a **Name** for the organization.
6. In the Organization Usages region, click **Add Row**.
7. From the **Usage** drop-down list, select **Service Organization**.
8. Click **Finish**.

Related Topics

- [How do I update existing setup data?](#)
- [Set Up an Offering with Scope](#)

Manage Resource Organization Hierarchies for Service BUs

After you define internal resource organizations to be associated with business units (BUs), you must add them 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. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Users and Security
 - o Task: Manage Resource Organization Hierarchies

2. In the Manage Resource Organization Hierarchies page, search for the resource organization that you created in the "Manage Internal Resource Organizations for Service BUs" procedure.
3. Click the link for the resource organization that you want to edit.
4. Select **Edit This Hierarchy Version** from the **Actions** menu.
5. Expand the organization list in the **Internal Resource Organization Hierarchy** region.
6. Select the organization that you created in the "Manage Internal Resource Organizations for Service BUs" procedure, to add it to the organization hierarchy.
7. Click **Add**.
8. In the **Add Tree Node** window, click **Search**.
9. In the **Search Node** window, search for the organization that you created in the **Manage Internal Resource Organizations** task.
10. Click **OK** to add the organization.

To add more organizations, select the parent node to add a child node. Repeat the steps to search and add as many times as needed.
11. Click **Save and Close**.
12. Click **Yes** on the warning message, which states that the hierarchy version is to be updated and the corresponding reporting hierarchy regenerated.

Related Topics

- [How do I update existing setup data?](#)
- [Set Up an Offering with Scope](#)

Create a Business Unit for Service

You can use a business unit (BU) to separate or share the following:

- Service setup data such as product catalog
- Transactional data such as service requests

To support the multiple-BU features, you must first create the required BUs.

To create a BU for the Service offering:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Company Profile
 - Task: Manage Business Unit
2. In the Manage Business Unit page, click **Create**.
3. In the Create Business Unit page, enter a name for the BU.
4. In the **Default Set** drop-down list, click **Search**.
5. In the **Reference Data Set Name** field, search for **Common**.
6. Select **COMMON** from the search results.
7. Click **OK**.
8. On the Create Business Unit page, click **Save and Close**.

To add another BU, select the **Manage Business Unit** task again and repeat the steps.

Related Topics

- [How do I update existing setup data?](#)
- [Set Up an Offering with Scope](#)

Associate Resource Organizations with Business Units

You can associate a resource organization with multiple business units (BUs). By associating resource organizations with BUs, you can control access to the transactional data available to service resources in business objects such as service requests.

Note: If you enable multiple BUs, ensure that each user is associated with a resource organization, and the resource organization is associated with one or more BUs. Otherwise, the default BU set on the HZ_DEFAULT_BU_CRM profile option is associated with SRs.

To associate a resource organization with business units:

1. Sign in to the application as an administrator or a setup user.
2. In the Navigator, select **Resource Directory**.
3. On the Resource Directory page, click the Tasks panel tab.
4. In the Resource Organizations region, click **View Organizations**.
5. On the View Organizations page, search for the organization that you created.
6. In the Search Results region, click the link for the organization.
7. Select the Business Units tab.
8. Click **Add Row**.
9. Select the business unit from the drop-down list.

Note: The first BU with which you associate the resource organization becomes the primary BU. If you associate the organization with more BUs, you can change the primary BU as required.

10. To add more business units, click **Save**, and then click **Add Row**.
11. After you add the business units, click **Save and Close**.
12. Click **Done**.

Create Employees for a Service BU

You can add resources to the resource organizations that you associate with business units (BUs). When you add a resource to an organization, the resource becomes a member of the organization and a part of the organization hierarchy.

You must first create a manager for the organization. To create application users, use the Users and Roles task:

1. Sign in to the application as an administrator or a setup user.
2. In the navigator, select **Users and Roles**.
3. On the Search Person page, click **Create**.
4. On the Create User page, enter the **Last Name** and **First Name**.

5. Enter the **Email**.
6. In the **User Details** region, enter a **User Name**.
7. In the **Employment Information** region, select **Employee** from the **Person Type** drop-down list.
8. Select a **Legal Employer** from the drop-down list.
9. Select the **Business Unit** of the employee from the drop-down list.

Note: This BU that you select is the BU of the employee, and not the BU of the resource organization. They both may be different. What BU you select for the employee information depends on how employees are organized.

10. In the **Resource Information** region, select the **Resource Role** from the drop-down list.
11. Search for and select an organization from the **Organization** drop-down list.

This organization is the one that you created earlier. The agent is associated with the BU through this organization.

12. Click **Autoprovision Roles**. This gives the user any predefined job roles.
13. Click **Save and Close**.

Repeat the steps to create another user who is the employee of the manager. The steps are the same except that in the Resource Information region, you search for and add the Reporting Manager that you already created.

14. When you have added all the users, click **Autoprovision Roles**.
15. Click **Save and Close**.

You can view everyone you created in the **Resource Directory** by using the Navigator.

View Employees in the Resource Directory

To view the employees that you created in the Resource Directory:

1. In the Navigator, click **Resource Directory**.
2. Click the Tasks panel tab.
3. In the Resource Organizations region, click **View Organizations**.
4. On the View Organizations page, in the **Organization** field, type the name of the organization for which you want to view the employees.
5. From the displayed list of organizations matching your search, click the organization that you want.

The list of employees is displayed in the Members tab.

For more information about the Resource Directory, see the *Oracle Fusion Automation Using Customer Data Management for Sales and Service* guide.

Related Topics

- [Overview of Setting Up Users and Security](#)
- [About Security Roles](#)
- [How do I enable desktop pages for Resource Directory?](#)

How do I set the scope in Service BU setup?

When you opt in to the business units (BUs) feature for Service, the Scope column in the Business Units tasks list contains links to set the scope of the task.

Let's learn how to set the scope for tasks when setting up BUs for Service.

To set the scope for tasks when you set up BUs for Service:

1. Sign in to the application as an administrator or a setup user.
2. Navigate to **Setup and Maintenance**.
3. From the **Setup** drop-down list, select the Service offering.
4. Select the **Business Units** functional area.
5. From the **Show** drop-down list, select **All Tasks**.
6. In the **Scope** column for any task in the task list, click the displayed link.

The **Select Scope** dialog box is displayed, and the task for which you are setting the scope is already selected.

Note: When you go to the task list for the first time, the scope may not be set and the **Scope** column displays the **Select** link. After you set the scope once, that BU appears as a link in the Scope column.

7. From the **Business Unit** drop-down list in the **Select Scope** dialog box, select **Select and Add**.
8. Click **Apply and Go to Task**.
9. On the Select and Add: Business Unit page, search for and select the BU that you want to set for the scope.
10. Click **Save and Close**.

The page opens for the task you're working with. On this page, you can choose to use the default **Site Level Value** or select the **Business Unit Profile Value**.

11. To select a Business Unit Profile Value:
 - a. Deselect the **Use Site Value** check box.
 - b. In the **Business Unit Profile Value** field, enter the profile value for the BU.
 - c. Click **Done**.

The task closes and now on the Setup page, the BU that you set for the Scope is populated for all tasks. Each task you open now is the setup for the BU in the **Scope** column.

To set up additional BUs, repeat the same steps.

Note:

- After you set up the first BU, the **Business Unit** drop-down list in the **Select Scope** window shows the BUs that you already set up.
- For all tasks, the scope displays the BU that you're currently working with. To change the BU again, click the BU in the **Scope** column for any of the tasks.

Overview of optional setup tasks for Service BU

Once the preliminary steps for setting up Service business units (BUs) are done, you can set up BU-specific tasks. All the BU-specific tasks are grouped in the Business Units functional area.

For multiple BUs in Service, the setup tasks are the same as in a normal Service setup. The only difference is that with most tasks for multiple BUs, you can do one of the following:

- Use the default site-value profile option.
- Set or select the business unit profile value.

However, the previous statement isn't true for some tasks for multiple BUs. The following multiple-BU tasks aren't related to profile options, so you must configure these tasks for each BU separately:

- Manage Service Categories for Business Units
- Manage HR Help Desk Service Categories for Business Units
- Manage Communication Channels for Business Units

Note: Categories and channels are restricted based on the BU set as the scope.

- When a new category or channel is created, it's automatically associated with the BU set in the scope. For more information about setting the scope for tasks when setting up BUs, see "Set the Scope in Service BU Setup".
- You can set the BU only for the top-level category. The BU on the child categories is automatically set based on the BU of the root category.

Here's a list of the other Service optional setup tasks and the help topics that provide more information.

Service Task	Related Topics
Manage Business Unit	Define Business Units
Manage Service Product Groups Usage for Business Unit Manage HR Help Desk Product Group Usage for Business Unit	Define a Catalog for the Service Offering Associate Different Catalogs to Different Business Units
Manage Service Categories for Business Unit Manage HR Help Desk Service Categories for Business Unit	Manage Service Request Categories Manage Service Request Categories and Product Usage Groups for HR Help Desk
Manage Communication Channels for Business Unit	Managing Communication Channels
Manage Service Email Templates for Business Unit	Define Email Templates Configure HR Help Desk Email

Service Task	Related Topics
Manage Human Resources Help Desk Email Templates for Business Unit	Note: You must perform extra steps to complete the Manage Service Email Templates for Business Unit task. In Functional Setup Manager, in the Business Units functional area of the Service offering, show All Tasks . For each task, click the Scope link and select the BU that you want. When you finish, click the Select and Go to Task button.
Manage Service Milestone Configuration for Business Unit	Overview of Milestones for Multiple BUs
Manage Outbound Email Profile Values for Business Unit	Configure Profile Options for Inbound and Outbound Email

Related Topics

- [How do I define a catalog for Service?](#)
- [How do I define email templates?](#)
- [How do I set the scope in Service BU setup?](#)
- [How do I manage categories?](#)

Secure the Business Unit Field on a Service Request

Let's say you want to restrict users with certain job roles from changing the business unit (BU) while editing a service request (SR). You can secure the BU field on the SR object by using the Update Service Request Business Unit privilege.

If you remove this privilege from the listed job roles, users with these roles can't change the BU of the SR.

The following predefined job roles can change the BU when they edit an SR:

- 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

Remove the Update Service Request Business Unit Privilege

If you remove the Update Service Request Business Unit privilege from the job roles mentioned in the previous list, users with those roles can't change the BU of the SR when editing.

Note: You can remove the Update Service Request Business Unit privilege from a user's job role only if you had granted one of the predefined service job roles from the previous list 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 predefined job role that's granted to the user. This predefined 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's 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 predefined 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

- [Guidelines for Copying Roles](#)

Export and Import the Functional Setup Data for Business Units

You may have created the functional setup data for all business units (BUs) in the Service offering in your test environment. After testing it, you may want to export the data to your production environment.

You can export or import the functional setup data by using the export and import feature in Functional Setup Manager. You can complete this process by using either 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 BUs. You can also export and import the functional setup data for a

specific BU by using an implementation project. This way, you have additional management flexibility in scenarios where each BU manages its own set of configurations.

For more information about the following features in Functional Setup Manager, see the Oracle Fusion Using Functional Setup Manager guide.

- Using implementation projects
- Export and import feature
- Implementation method based on offering

To export the functional setup data for BUs in the Service offering by using an implementation project:

1. Sign in as a setup user.
2. Navigate to the **Setup and Maintenance** work area and open the Tasks panel tab.
3. Select **Manage Implementation Projects**.
4. On the Manage Implementation Projects page, create a new implementation project to export the functional setup data for one of the following:
 - A specific business unit
 - All business units
5. Save and open the implementation project.
6. In the **Task Lists and Tasks** area, ensure that you add the **Define Business Units for Service** task list to the project.
7. Determine whether you want to export the setup data for a specific BU or all BUs.
 - If you want to export the setup data for all BUs, then go to step 12.
 - If you want to export the setup data for a specific BU, then complete all the remaining steps.
8. Expand the task list.
9. Navigate to the first task that has a **Select** link in the **Selected Scope** column. Click the **Select** link.
10. From the **Business Unit** list, select **Select and Add**.
11. Click **Apply and Go to Task**.
12. In the **Select and Add Business Unit** window, search and add the specific BU.
The selected BU appears in the **Selected Scope** column for all the tasks in the task list.
13. Click **Done** for the implementation project.
14. Continue to create the configuration package as described in the "Exporting Setup Data Using Implementation Project" process in the Oracle Fusion Using Functional Setup Manager guide.

When you export the configuration package:

- If you select all BUs in step 3, the export data includes the functional setup data for all BUs in the Service offering.
- If you select a specific BU in step 3, the export data includes the functional setup data only for the selected BU.

Note: To import the functional setup data for BUs, follow the steps described in the *Using Functional Setup Manager* guide.

Related Topics

- [Setup Data Export and Import Using Implementation Project](#)
- [Export Setup Data Using Implementation Project](#)
- [Setup Data Export and Import Using an Offering or a Functional Area](#)

Overview of BU-Based Service Request Visibility

When users view lists of service requests (SRs) or create user-defined searches, they can typically see all SRs. But you can restrict their access based on their business unit (BU) membership, if required. The predefined roles don't have SR visibility based on BU.

In the Security Console, you can assign BU-based visibility for SRs to specific roles. Consequently, users with these specific roles can see only the SRs assigned to the BUs where they're a resource member.

With this data security policy, your company can ensure that all predefined and user-defined searches are restricted only to BU membership for a set of users.

For more information about assigning the data security policies based on BU to your users, see [Set Up Service Request Visibility Based on BU](#).

Related Topics

- [How You Set Up Visibility Based on BU](#)

Migrate Business Objects Between Different Business Units

Sometimes, you may want to migrate some business objects between business units (BUs). But not all objects can be migrated.

You can migrate only the following business objects from one BU to another:

- Category
- Channel
- Milestone Configuration
- Service Request

Note: You can migrate only one business object at a time.

To migrate the listed business objects from one BU to another:

1. In the Navigator, click **Scheduled Processes**.
2. On the Overview page, click **Schedule New Process**.
3. In the **Schedule New Process** dialog box, select **Job** as the **Type** option.
4. Click the **Name** drop-down list and click **Search** at the end of the displayed list.
5. In the **Search and Select: Name** dialog box, type **Migrate** and click **Search**.
6. From the displayed list of jobs, select **Migrate Service Business Unit Data** and click **OK**.
The **Schedule New Process** dialog box displays the name of the selected job.
7. Click **OK**.

8. In the **Business Object** field in the **Process Details** dialog box, select the business object that you want to migrate.
9. From the **From Business Unit** drop-down list, select the BU from which you want to migrate the business object.

Note: If some existing records aren't associated with any BU, you can leave the **From Business Unit** field blank.
10. From the **To Business Unit** drop-down list, select the BU to which you're migrating the business object.
11. Click **Submit**.

Associate Different Catalogs with Different Business Units

When you have multiple BUs, you must create multiple usages, one for each BU. You can then associate a different catalog with each usage. This way, each BU is associated with a separate catalog.

If you have two BUs and you want to associate two different catalogs with the two BUs:

1. Create two different catalogs.
For more information about creating catalogs, see "Sales Catalog High-Level Setup Steps".
2. Create two usages for the two BUs.
 - a. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Service Catalog
 - Task: Manage Product Group Usage
 - b. In the Manage Product Group Usage page, go to the Product Group Usage region.
 - c. Click the **Actions** drop-down list.
 - d. Click **Create**.
 - e. In the **Create Product Group Usage** dialog box, in the **Usage** field, specify a name. For example, **BU_1_Usage**.
 - f. In the **Usage Code** field, specify a code.
 - g. Deselect the **Allow Duplicate Content** check box.
 - h. Click **OK**.
 - i. To create a second usage for the second BU, repeat steps c to h.
3. Associate a catalog with a usage.
For more information about associating a catalog with a usage, see "Define a Catalog for the Service Offering".
4. Associate a BU with a usage.
 - a. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Business Units
 - Task: Manage Service Product Group Usage for Business Unit
 - b. For the first BU, deselect the **Use Site Value** check box.

-
- c. In the **Business Unit Profile Value** column, specify the usage code of the first usage that you created.

Note: You must specify the usage code, and not the usage name.

5. Click **Done**.
6. Repeat steps 3 to 5 for the second usage and the second BU.

Related Topics

- [Sales Catalog High-Level Setup Steps](#)
- [How do I define a catalog for Service?](#)

12 Import and Export

What's Import and Export Management?

Import and Export Management is used to import and export a wide range of application data using import and export management in your service application.

For example, you can use Import and Export management to export object data so that you can 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.

Overview of Import Objects

You can import several objects into the application using the import management framework.

To import objects, Click **Tools > Import Management > Import Objects**.

The following is a sample list of top-level and sublevel objects that you can import into the service application. For more information on these and other objects available for import, see the Oracle Fusion Understanding Import and Export Management for Sales and Service guide.

- Service Requests
 - Messages
 - Contact Members
 - Resource Members
 - References
- Queues
 - Queue Party Resources
 - Queue Team Resources

- Categories

Note:

- Since the service request data depends on queues and categories, you must import the queues and categories before importing your service requests.
- Interactions: You can't import interactions into the application, 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

Overview of Export Objects

You can extract large volumes of data from the application using bulk export. You can either extract a full set of records for an object, or perform incremental extracts. For example, you can extract the complete set of service requests data or extract an updated set of records every week.

To export objects, Click **Tools > Export Management > Export Objects**. For more information, see the Oracle Fusion Understanding Import and Export Management for Sales and Service guide.

Use Functional Setup Manager to Import, Export, and Compare Categories Between Environments

Your users can import, export, and compare service categories between environments using Functional Setup Manager. Users to create or update setup service category data in bulk as well as compare differences between two environments.

For example, a user wants to import service category data into a production environment that has been exported from a test environment after making changes to the setup data. Before the user imports the data, the user can compare the exported data of the test environment with the existing production data, and preview how the setup data in production will change after the data is imported.

Any implementation usually requires migrating setup data from one environment to another at various points in the subscription lifecycle. For example, a subscribed offering is typically set up in the test environment first, and is moved to the production environment only after proper testing and verification. Setup export and import processes help you migrate setup data from test to production.

Note: You must have the Application Implementation Consultant role (ORA_ASM_APPLICATION_IMPLEMENTATION_CONSULTANT_JOB) to export, import, and compare setup data.

For more information, refer to the Related Topics section for a link to the Oracle Fusion Using Functional Setup Manager guide.

Related Topics

- [Export and Import CSV File Packages](#)

Before You Import

Since the service request data is dependent upon queues and categories, you must import the queues and categories before importing your service requests. 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 "Import Your Account Data" in the guide specified in Related Topics.
- **Contacts:** The list of contacts for each account that can be associated with the SR. For information about importing contacts, see "Import Your Contact Data" in the guide specified in Related Topics.
- **Resources:** The list of resources to which you can assign work objects in the SR. For information about importing resources, see "Import Resource Data" in the guide specified in Related Topics.
- **Partners:** The list of partner accounts that you can associate with an SR. For information about importing partners, see "Import Your Partner Data" in the guide specified in Related Topics.
- **Product Groups:** A product group is a group of related products. For information about importing product groups, see "Import Your Product Group Data" in the guide specified in Related Topics.
- **Products:** List of products against which you can raise an SR. For information about importing products, see "Import Your Product Data" in the guide specified in in Related Topics.
- **Asset:** List of assets associated with the account. For information about importing assets, see "Import Your Asset Data" in the guide specified in Related Topics.
- **Categories:** The categories associated with the SRs. For information about importing categories, see "Import Your Service Request Data" in the guide specified in Related Topics.
- **Queues:** The queues to which your SRs are assigned. For information about importing queues, see "Import Your Queue Data" in the guide specified in Related Topics.
- **Channels:** Import the channels that are already associated with your SRs. For information about importing channels, see "Import Your Service Request Data" in the guide specified in Related Topics.
- **Tags:** The administrator-defined tags to be associated with SRs. For information about importing tags, see "Import Your Service Request Data" in the guide specified in Related Topics.

Related Topics

- [Oracle Fusion Understanding Import and Export Management for Sales and Service](#)

Import Objects

After you import the dependent objects, create an import activity to upload details of objects you want to upload. Here's how you import an object:

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 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. 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 Mapping page.

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.

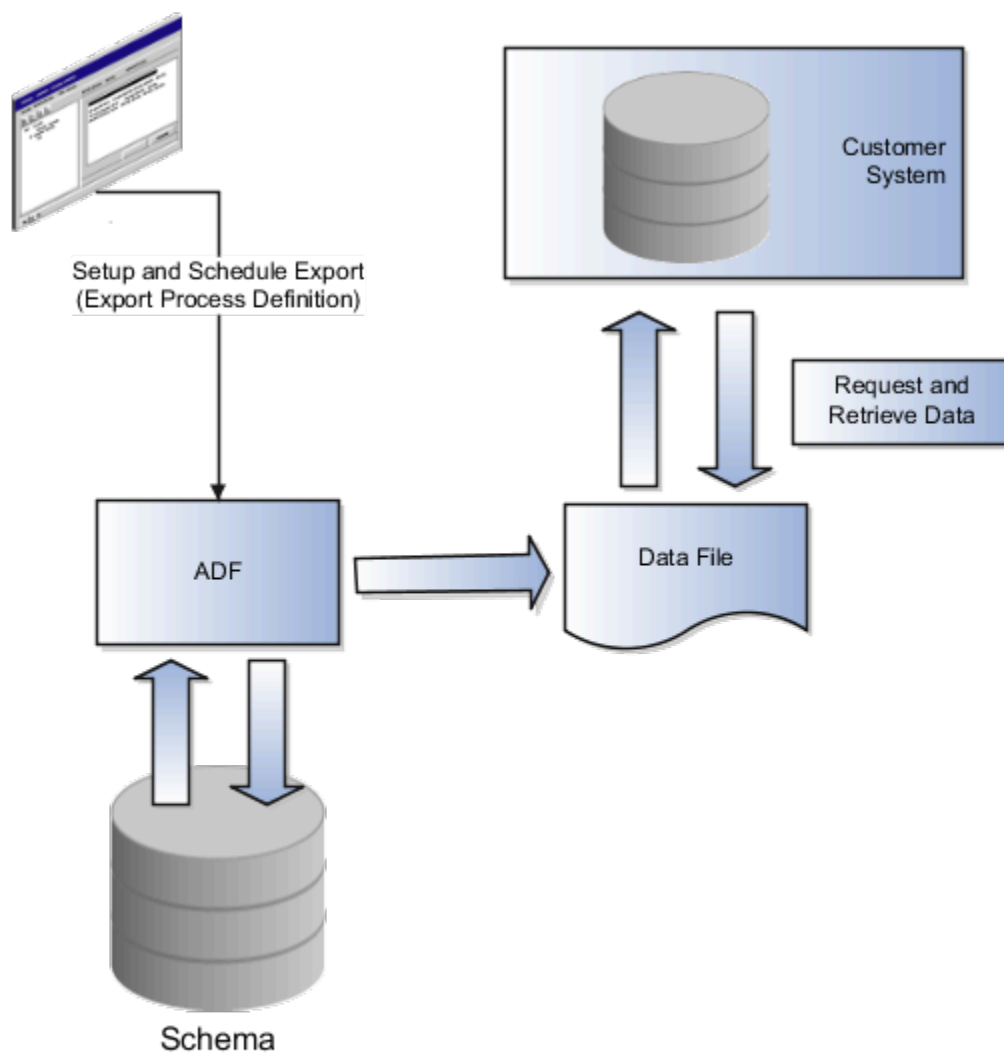
Review your import details and click **Activate** to activate the import activity.

You can use predefined .csv templates for your requests, queues, and categories that are provided in the application. You can also import user-defined objects.

Overview of Bulk Data Export

You can extract large volumes of data 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.



13 Real-Time Channels

Chat

How do I enable features and profile options for chat?

You might need to enable the Omnichannel Routing and Chat features to access the profile options.

Enable Omnichannel Routing and Chat features

To ensure your features are enabled, A user with administration privileges should perform the following steps:

1. Navigate to Setup and Maintenance, and select **Service** from the Setup offering page.
2. Click the **Change Feature Opt In** link.
3. Highlight the **Communication Channels** row and click the pencil icon to access the edit view and edit your features.
4. If the Omnichannel Routing feature isn't enabled, check the **Enable** field.
5. If the Chat feature isn't enabled, check the **Enable** field.
6. Click **Done**.

Enable Omnichannel profile options

Once you've confirmed that the Omnichannel and Chat features are enabled, you'll need to configure the following profile options to enable various chat features.

1. In Functional Setup Manager, select **Communication Channels** from the Service offering.
2. In the **Show** drop-down list, select **All Tasks** to ensure all Tasks are displayed.
3. Select the **Manage Omnichannel Profile Options** task.
4. View and edit Omnichannel profile options as needed:

Omnichannel Profile options

Profile Option	Description	Suggested value
SVC_ENABLE_OMNI	Enable the sending of work requests to Omnichannel for assignment to an agent.	Yes
SVC_OMNICHANNEL_BROWSER_NOTIFICATION_ENA	Used to enable real-time browser notifications.	Yes
SVC_OMNICHANNEL_DESKTOP_NOTIFICATION_ENA	Enable toast messages to view desktop notifications even when you're not actively viewing the application pages.	Yes
SVC_OMNI_DISABLE_WORK_ITEM_AVAILABILITY_SELECTION	Disables the option for agents to change their availability to handle work items on the Omnichannel toolbar.	None All

Profile Option	Description	Suggested value
		Chat Non-Real-Time Work

Enable chat profile options

1. In Functional Setup Manager, select **Communication Channels** from the Service offering.
2. In the **Show** drop-down list, select **All Tasks** to ensure all Tasks are displayed.
3. Select the **Manage Chat Profile Options** task.
4. View and edit Chat profile options as needed:

Chat profile options

Profile Option	Description	Suggested value
SVC_ENABLE_CHAT	Enable the Chat feature to make users available for chat and allow them to receive incoming chat requests.	Yes
SVC_CHAT_OFFER_NOTIFICATION_TIMEOUT_VALUE	Specify the seconds that lapse before the chat offer notification is automatically dismissed. The default number is 30 but the dismiss interval can be set between 5 and 285 seconds.	30
SVC_CHAT_TRANSFER_TO_QUEUE_ENABLED	Enable transferring to queue for the Chat agent.	Yes
ORA_SVC_CHAT_TRANSFER_TO_AGENT_ENABLED	Enable transferring to of a chat to another chat agent.	Yes
ORA_SVC_CHAT_CONFERENCE_WITH_AGENT_ENABLED	Enable a chat conference between agents.	Yes
SVC_CHAT_ANONYMOUS_ACCESS_ENABLED	Enable chat anonymous access for the end users	Yes Note: Set Anonymous Access to No if chat is only used for the Help Desk. If it's being used for other applications, such as Service Center, then it should be set to Yes.
SVC_CHAT_INLAYS_ACCESS_ENABLED	Enable access for chat inlays to get the bootstrap configurations.	Yes For more information, see <i>How do I implement Embedded Service?</i>

Profile Option	Description	Suggested value
SVC_CHAT_CKEDITOR_ENABLE	Enable the CK Editor for the Chat agent.	Yes

Enable MCA profile options

1. In Functional Setup Manager, select **Communication Channels** from the Service offering.
2. In the **Show** drop-down list, select **All Tasks** to ensure all Tasks are displayed.
3. Select the **Manage MCA Profile Options** task.
4. View and edit MCA profile options as needed:

MCA profile options

Profile Option	Description	Suggested value
SVC_MCA_SHOW_CONTACT_ON_WRAP_UP	Indicates whether to show the contact information during the wrap up process.	Yes

Configure Call Flow Parameters

Use the Configure Call Flow Parameters task to specify whether a contact must be identified and validated or if a screen pop must be displayed. You can also use these settings to specify when and how a call wrap up must be displayed.

1. In Functional Setup Manager, select **Communication Channels** from the Service offering.
2. In the **Show** drop-down list, select **All Tasks** to ensure all Tasks are displayed.
3. Select the **Configure Call Flow Parameters** task.
4. View and edit MCA profile options as needed:

Call Flow Parameter option values

Profile Option	Description	Suggestion value
Contact Required	Used to require that a chat must have a contact.	Yes
Contact Verification	Used to require that a contact of a chat must be verified.	Yes
Screen Pop	Used to screen pop the appropriate page.	Yes
Wrap Up	Used to present a Wrap Up page to the agent at the end of a chat.	Yes Note: Oracle strongly recommends enabling wrap ups for chats if you're using Service Center to prevent the transcript from closing at the end of a chat.

Configure optional profile options

Now, you can review or change profile options accessed by performing the following steps. Note that these usually don't need to be changed.

1. Click the **Tasks** icon in Functional Setup Manager and select **Search**.
2. Type: **Manage Global Search Profile Options** and click **Search**.
3. Select the **Manage Global Search Profile Options** task.
4. View and configure the profile options listed in the following table:

Chat profile options

Profile Option	Description	Suggested value
SVC_OMNI_REINSTATE_WORK_ITEM_AVAILABILITY	When returning to the supported modules, it automatically reinstates the non-real-time channel availability groups for work Items on the Omnichannel toolbar.	• None • None-Real-Time Work
SVC_CHAT_IDLE_TIMEOUT	The number of seconds after which a chat will be terminated if there's no activity. Zero disables the functionality. Except zero, the minimum value is 300. Default is 600.	600
ORA_SVC_OMNI_BROWSER_INACTIVITY_TIMEOUT	Profile option that allows administrators to set ADS timeout due to inactivity; this can be increased up to 8 hours (480). Change the number of minutes of browser inactivity before work assignment offers will end. The minimum default value is 30 minutes, and the maximum value can be up to 480 minutes.	30
ORA_SVC_OMNI_MARK_UNAVAILABLE_ON_BROWSER_INACTIVITY	Enables the system to mark agent unavailable for work after browser inactivity for the duration specified by ORA_SVC_OMNI_BROWSER_INACTIVITY_TIMEOUT.	No
ORA_SVC_QUEUED_REQUEST_DISCARD_SECONDS	Specify the queue time if no agents are available. If the value is 0, the chat will stay queued until the customer ends the request. The minimum is 120 and the maximum is 1800.	1800
ORA_SVC_OMNI_AGENT_SESSIONS_COUNT_INTERVAL	Change the interval number of minutes to look for other active browser sessions before marking a user offline. The maximum value can be up to 480 minutes.	60
ORA_SVC_MCA_GLOBAL_PRESENCE_ENABLED	Decides whether a customer has opted in for global presence across the application.	Yes

Profile Option	Description	Suggested value
		Note: You can set this profile option at the site level.

How do I enable and set up chat?

To enable the service representatives to use the chat function and to receive chat notifications, you must complete the following steps.

1. Opt into the Omnichannel Routing feature
2. Opt into the Chat feature
3. Enable Omnichannel profile option
4. Enable notifications
5. Enable Chat profile option
6. Enable the Chat Inlays profile option
7. Enable the Anonymous Access profile option
8. Provide the Chat Agent role to users who will be handling chats
9. Create your queues and associate your chat agents
10. Create your assignment rules to route your chats
11. Set up your Digital Customer Service users

For more information about enabling Omnichannel to route your chats, see [How do I automatically route chat work assignments to agents?](#)

Configure Chat Wrap Up

You wrap up a chat, collecting all relevant information from an interaction, in the Chat Wrap Up window. You access past wrap up records in the Interaction History view.

If a service request, contact, or account is associated with a chat interaction, the wrap up, along with a link to the chat transcript, is written to the record. This doesn't happen unless wrap up is enabled. Chat wrap up is enabled by default.

An agent workflow for a chat interaction includes the option to display a wrap up dialog box at the close of an interaction. The workflow captures the full chat transcript between the agent and consumer and associates it with the interaction wrap-up record.

Note: The chat transcript is linked to an SR only when the contact in the chat interaction is associated with an account, and they're the primary contact on the SR. When a chat is started for a contact that doesn't have an account linked to it, the chat transcript is linked to the contact, but not to the SR. For more information, see [Overview of Cross-Channel Interactions](#).

You can view details of the chat transcript at any time in the future from the Interaction History tab or within an account, contact and service request record by drilling down into the associated wrap-up record available on the chat interaction. If the chat transcript has been disabled, here's how to enable it.

1. As an Administrator, go into Setup and Maintenance.
2. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Configure Call Flow Parameters
3. Click the **Wrap Up** drop-down list and select **Yes**. This is the default value for this setting.

For more information on other call flow parameters, see Related Links for the Configure Call Flow Parameter topic.

After the chat exchange is finished, the agent closes the chat within Live Window and then enters the associated interaction includes wrap-up details and the transcript.

Overview of Extending the Wrap Up UI

1. Open the Oracle Cloud Application page containing the component you want to extend.
2. Click the **Settings and Actions** menu and select **Edit Page in Visual Builder** to open the page in the **VB Studio Page Designer**.
3. Create a project or Open project with base application as: **Oracle CX Service UI Extension APP**.
4. Click the **Layouts** sub tab, then from the Dependencies list click **WrapUps**.
5. In the Dynamic Form list, select the **Wrapup dynamic layout**.
6. In the **Rule Set** field, click the **Parameters** tab.
7. The Wrapup component has the following three contexts:
 - o appClassification
 - o callDirection
 - o channelType
8. Mouse hover each context to highlight it, then hover of the **?** symbol to view possible values for each component context:
 - o appClassification: Possible Values: ORA_SALES, ORA_SERVICE, ORA_HRHD
 - o callDirection: Possible Values: ORA_SVC_INBOUND, ORA_SVC_OUTBOUND
 - o channelType: Possible Values: ORA_SVC_PHONE, ORA_SVC_CHAT
9. In the **Display Logic** area, click the **Duplicate Rule** icon to clone the default rule.
10. Enter a name for the duplicate rule then click the **Duplicate** button to create a copy of rule and layout.
Note: Default layouts from dependencies are read-only, you can't edit them.
11. In the newly copied layout, select **Click to add condition** to add conditions for when to use this rule and layout.
12. You can add conditions based on the following:
 - o Context
 - o Fields
 - o Responsive
 - o User
13. You can give multiple conditions, or you can add multiple rules based on your requirement.
14. Now we'll create rules, so go to the Layouts area, click the **+ Rule** button. For more information on create a rule, see *Create a Rule in a Rule Set*.

Further Extending You Can Do

For further information on extending the application, see these topics:

- To add more fields to the layout, see [How do I add, hide, or rearrange fields?](#).
- To delete a field, hover over the field, and click the Delete button. <insert graphic>
- To mark a field as required, click the Required checkbox in the properties for the given field. <insert graphic>
- To make fields conditionally required or read only, see [How do I make a field read only or required?](#).

Related Topics

- [Overview of Cross-Channel Interactions](#)

Enable Deletion of Chat Transcripts to Comply with General Data Protection Regulation (GDPR)

Customers in the European Union (EU) can request the deletion of their chat transcripts under General Data Protection Regulation (GDPR) regulations.

The GDPR is a regulation in EU law. This law governs data protection and privacy for all individual citizens of the EU and the European Economic Area (EEA). One of the GDPR regulations is the right to erasure or the right to be forgotten. This means that your customer may want to request the deletion of all chat conversations for various reasons related to data protection and privacy. You can configure chat in your service application to purge the chat transcripts, either periodically by setting profile options or manually upon your customer's request.

Here's what you must do to enable purging of chat transcripts:

- The SVC_DELETE_CHAT_TRANSCRIPT_PRIV privilege lets users manually delete chat transcripts from the user interface. By default, this permission is given only to the Customer Service Administrator role. To permit other users or roles to delete chat transcripts, you must add this privilege to the respective roles.
- To automatically delete chat transcripts periodically, you must also set the following profile options:
 - SVC_CHAT_TRANSCRIPT_PURGE_ENABLED
 - SVC_CHAT_TRANSCRIPT_PURGE_INTERVAL
 - SVC_CHAT_TRANSCRIPT_DELETE_ENABLED
 - SVC_CHAT_TRANSCRIPT_PURGE_ITERATION_INTERVAL
 - SVC_CHAT_TRANSCRIPT_PURGE_BATCH_SIZE

Note: For more information about how to set the profile options, see "Configure Chat Profile Options".

Note: For more information about how to delete chat transcripts manually, see "Delete Chat Transcripts to Comply with General Data Protection Regulation (GDPR)".

Related Topics

- [How do I enable features and profile options for chat?](#)
- [Delete Chat Transcripts to Comply with General Data Protection Regulation \(GDPR\)](#)

Show chat and phone call panels based on custom objects

You can extend your Service application to render the chat and phone call panels in UI pages based on custom objects.

This allows your agents to interact with customers using these UIs. Agents can take notes while working alongside the UIs you create. If needed, administrators can configure the application to screen pop to the custom UI.

Use Visual Builder Studio to expose your UIs in the application.

To render the Chat or Phone Call panels in pages you create, you need to add the fragment `cx-svc-mca-communicator` to the page with required input parameters.

The following table shows the properties of the `cx-svc-mca-communicator` fragment.

Name	Description	Default value
appExtensionId	App Extension ID of the base application.	oracle_cx_serviceUI
chatHeaderLayout	For changing the default chat header layout to a new layout.	chatHeader
chatHeaderMetadata	Chat header metadata provider.	{ "type": "vb/ DynamicLayoutMetadataProviderDescriptor", "defaultValue": { "endpoint": "oracle_cx_ serviceUI:contacts/get_contacts" } }
contactVerificationLayout	For changing the default contact verification layout to a new layout.	contactVerify
ctiHeaderContext	CTI header context for changing the call header layout to a new one.	ctiHeader
mcaRestApiGroupNames	MCA Rest API group names used by MCA components.	NA
msiContext	MSI context.	{}
optInFeatures	List of opt-in features that should be enabled. For example, for enabling Agent Assist, recommended actions and summarization set the value as ["agent_assist", "recommended_actions", "summarization"].	[]

Name	Description	Default value
wrapupMetadata	Wrap up metadata provider.	{ "type": "vb/ DynamicLayoutMetadataProviderDescriptor", "defaultValue": { "endpoint": "oracle_cx_ serviceUI:wrapUps/create_wrapUps" } }
wrapUpReadonlyMetadata	Wrap up read only metadata provider.	{ "type": "vb/ DynamicLayoutMetadataProviderDescriptor", "defaultValue": { "endpoint": "oracle_cx_ serviceUI:wrapUps/get_wrapUps" } }

The fragment also handles the `broadcastChannel` messages and performs the required actions. For example, resolving the external URL for the KM article, opening the KM article from snippets, and propagating other events from the `mca-communication-wrapper` to the custom pages.

The following tables shows the events that are triggered from the fragment to the container:

mcaEngagementAddedEvent	Triggered when a Chat Offer or Call Notification is accepted.
mcaEngagementRemovedEvent	Triggered when engagement has been removed.
mcaEngagementStateChangeEvent	Triggered when the engagement state changes.
mcaEngagementStartWrapupEvent	Triggered when the engagement has transitioned to the Wrap Up phase.
mcaEngagementWrapupEndedEvent	Triggered when the engagement has completed.
kmArticleOpenEvent	Triggered when agent opens a KM article from suggestions.
kmArticleShareToChatEvent	Triggered when an agent shares a suggestion in a chat.
kmArticleShareToEmailEvent	Triggered when an agent shares a suggestion to email.
mcaRecommendationEvent	Triggered when an agent performs a recommended action.
mcaDelegatedRecommendationEvent	Triggered to check if the recommendation is delegated.

Required changes for implementing custom page screenpop in MSI mode

1. Create an app UI and update the page JSON to support embedded navigation:

```
"navigation": { "fromExternal": "enabled", "embeddable": "enabled" }
```

2. Add following metadata:

```
"contactsGetContactsMetadata": {
```

```
"type" : "vb/DynamicLayoutMetadataProviderDescriptor",  
"defaultValue" : {  
  "endpoint" : "oracle_cx_serviceUI:contacts/get_contacts"  
}  
}
```

3. Add the fragment `cx-svc-mca-communicator` to the page.

Steps to configure the profile option to enable Presence in the Global Header

To enable this functionality Presence must be enabled in the Global Header. Set the profile option

`ORA_SVC_MCA_GLOBAL_PRESENCE_ENABLED` to Yes, and Global Presence will be enabled. You can set this profile option either at a Site level or at a User level.

1. In Setup and Maintenance, click the Tasks icon, then click **Search**.
2. Search and select **Manage Administrator Profile Values**.
3. Enter `ORA_SVC_MCA_GLOBAL_PRESENCE_ENABLED` in the Profile Option Code field.
4. In the Profile Values section, do one of the following:
 - Set the Profile Value to **Yes** if you want all users to be able to set their presence from the global header.
 - Set the Site Level to **No** and add a new row for each user who would need to set their presence from the global header.
5. Click **Save and Close**.

Overview of Web Chat for Service

Web Chat can combine assisted service with a live agent and a self-service chatbot in a single client for a consistent end-user experience across channels.

Web Chat handles chats that can be directly routed and handled by a human agent as a more modern and richer alternative than using the previous Chat Inlay deployment. If you are licensed to use Oracle Digital Assistant, you can use that same Web Chat client with DA as an Agent. This treats your intelligent chatbot as a natural extension to service.

Note: Oracle Digital Assistant is not required, but an optional add-on.

Web Chat for Service enables rich conversational and intelligent assisted service experiences. It's a chat widget that combines native chat capabilities with the rich configuration and branding options of Oracle Digital Assistant (ODA) Web SDK into a single widget.

This means that a lot of the parameters you can set for your web chat experience are based on the Oracle Digital Assistant Native Client SDK for Web which makes it a rich experience. Web Chat for Service makes use of a chat session that occurs between an end user and a Chat Agent.

The Chat Agent can also be an Oracle Digital Assistant using the Digital Assistant as an Agent feature. The routing logic for when a chat needs to go to a bot or a live agent is defined in the Fusion Service chat assignment rules. Digital Assistant is treated as any other human agent and is assigned to a digital assistant queue. The benefit of this Digital Assistant as an Agent is the centralized setup and reporting across queues with a warm handover to a live agent. This seamless transfer with full history in the chat transcript guarantees a full 360 view for agents inside the Fusion Service Center. Seamless transfer from the bot to a live Chat Agent can be configured in your Fusion Service deployment and Digital Assistant Skill.

Web Chat for Service is a syndicated widget that you can deploy on your Digital Customer Service application or any web page through the Oracle Engagement Engine single tag. You configure your Web Chat Client in Engagement

Engine by creating the web chat component. You can define the desired functionality, layout, colors, icons etc in a very convenient and comprehensive way. Note that only a subset of the properties available in ODA Native Client SDK for Web are supported in the Web Chat Client.

Oracle Engagement Engine, is a product that enables an organization to unify rule definition and management into one scalable and responsive solution. This point-and click-interface allows for higher business agility to respond faster to changing customer journeys and makes these web journeys more engaging and personalized.

The primary tools of Engagement Engine are the Oracle Engagement Engine Rules Editor (Rules Editor) and the Oracle Engagement Engine Runtime. The main goal of the Rules Editor is to simplify the process of designing, testing, deploying, optimizing and measuring the impact of using rules to assist your business solution. The Runtime is used when you publish your rules from the Rules Editor.

You use the application to create, edit and delete rules, and deploy rules to sites. In addition to enabling rule definition, you use the application to manage a multi-site environment. A site is a means of organizing rules with a common context. Engagement Engine enables you to create and edit an unlimited number of sites.

A multi-site environment has a number of uses:

- Use it to deploy rules to internal sites for testing and analysis before they are deployed on any public-facing environment.
- Use it to distinguish between public-facing sites depending on specified criteria. In other words, you can define different sites for different product ranges, or for different geographical locations.

In case your Engagement Engine account has not been provisioned, you can request a new account by logging a Service Request with Oracle Support.

For more information on Oracle Engagement Engine, refer to [Administering Oracle Engagement Engine](#).

To learn more about Oracle Digital Assistant and Digital Assistant as an Agent, refer to: [Oracle Digital Assistant Get Started](#).

For information on setting up Digital Assistant as an Agent, refer to the Service Integration chapter in [Using Oracle Digital Assistant](#).

The attributes supported in the Web Chat Client through Engagement Engine are based on the properties that can be set in the [Oracle Digital Assistant Native Client SDK for Web](#). Note that only a subset of the properties available in ODA Native Client SDK for Web are supported in the Web Chat Client.

For setting up Digital Assistant as an Agent in your Fusion Service environment, you can find the setup steps in the following guides:

- [Implementing Service Center with the Classic User Experience](#) (Refer to User Oracle Digital Assistant as an Agent)
- [Implementing Service Center with the Redwood User Experience](#) (Refer to User Oracle Digital Assistant as an Agent)

What's the Digital Customer Service design time and runtime architecture?

Oracle Digital Customer Service is an offering within Oracle Fusion Service that lets you provide your customer account users self-service access to their service requests and relevant knowledge articles through a web interface.

You can configure the Digital Customer Service application user interface to reflect a company brand. Using Oracle Visual Builder, you define root pages and styles, and include various UI components, depending on your business needs.

Note: To use Chat inlays in Oracle Fusion Service, you must configure some profile options. For more information, see "Configure Chat Inlay" in *Oracle Fusion Service Implementing Digital Customer Service*.

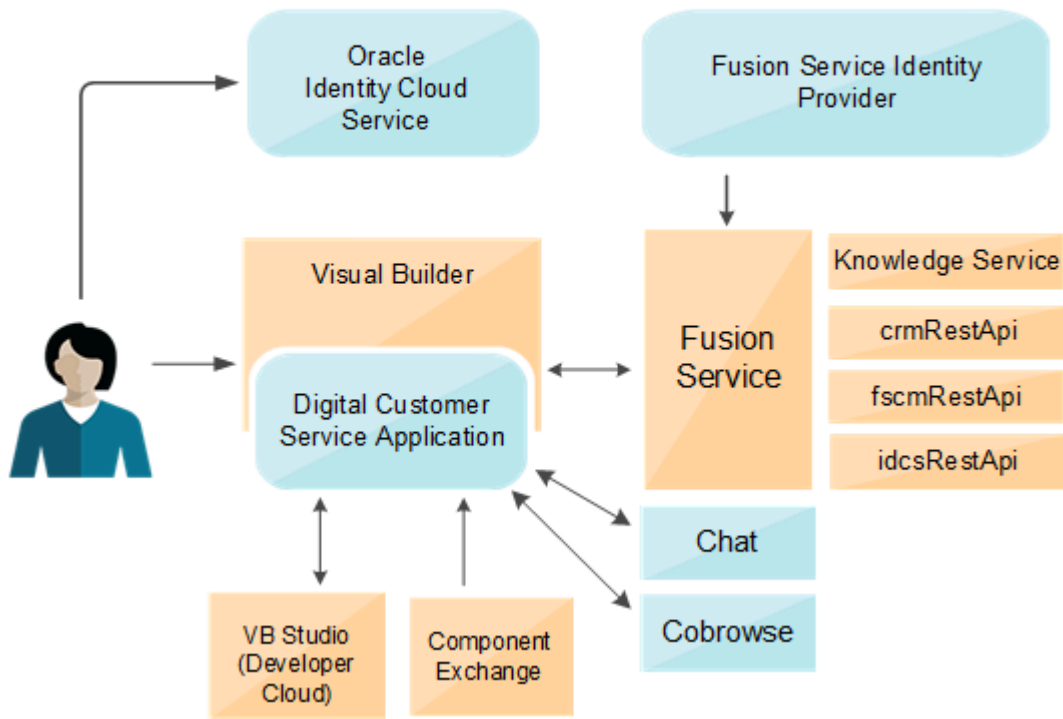
Once configured and deployed, your customers can self-serve through the application and search for knowledge articles to solve their problems. Also, your customers can register as Digital Customer Service self-service users so they can interact with your customer service representatives through service requests, work orders, or by chatting.

Here are general overviews of the design time and of the runtime architecture. The Digital Customer Service application relies on the following APIs:

- **crmRestApi.** Provides the connection to Fusion Service.
- **fscmRestApi.** Provides the connection to the Oracle Application Cloud topology manager and functional setup.
- **idcsRestApi.** Provides the connection to Oracle Identity Cloud Service.
- **knowledge-service.** Provides the connection to knowledge content and search.
- **attachmentDocTrackerRestApi.** Used by the file attachment upload mechanism.
- **attachmentUploadRestApi.** Used by the file attachment upload mechanism.
- **kmRestApi.** Used to access the Knowledge Management v2 REST API
- **utilityRestApi.** Used by ODCS Knowledge Attachment download mechanism.

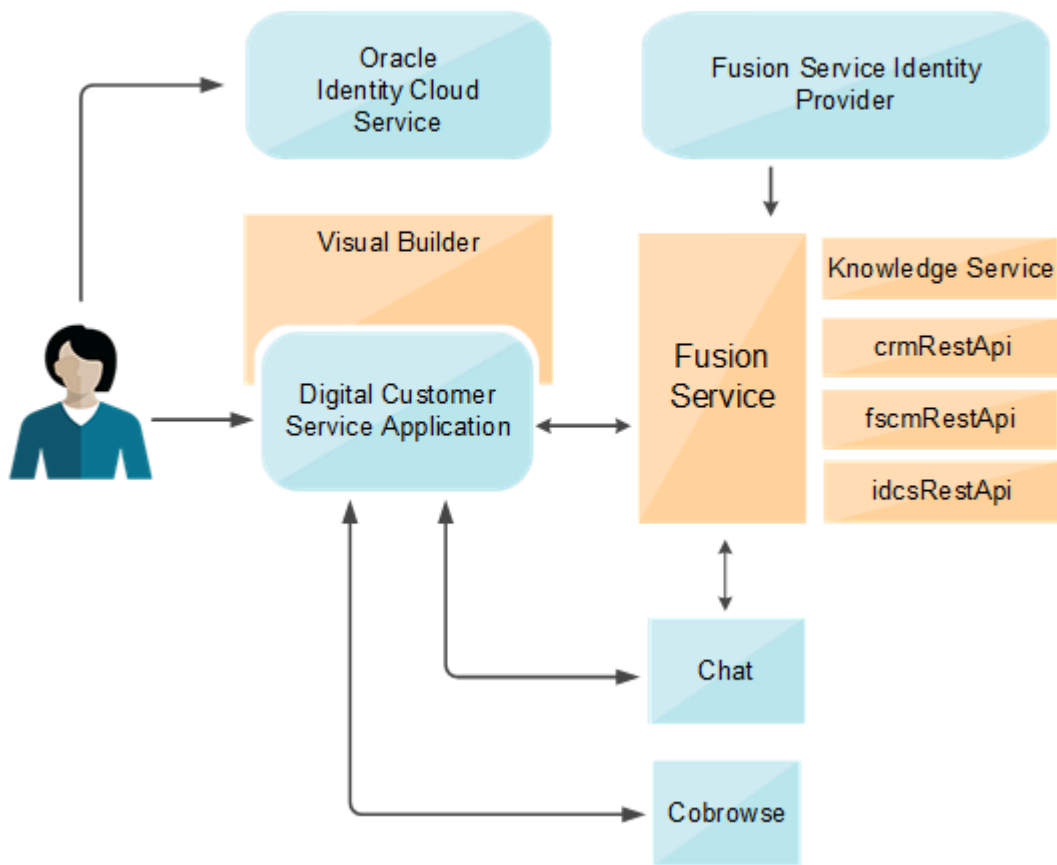
At design time, when developing a Digital Customer Service application, the developer logs into Oracle Identity Cloud Service to access the Visual Builder Designer and selects available application templates and components from the Component Exchange. The source code for the application can be maintained in a Git repository provided through the Visual Builder Studio (formerly known as Developer Cloud Service). The application is built to interact with various REST services from the associated Fusion Service instance. Here's a look at the design time architecture:

Digital Customer Service Design Time Architecture



At runtime, a self service user interacts with the Digital Customer Service application that's served up from the Visual Builder runtime environment, and can make requests for a self-service account that's created for the user on Oracle Identity Cloud Service. The user can then sign in to Oracle Identity Cloud Service to get authenticated access to the application and use the full functionality of the application enabled by the various REST services from Fusion Service. The application might also expose Chat functionality. Here's a look at the runtime architecture:

Digital Customer Service Run Time Architecture



Computer Telephony Integration (CTI)

What's Computer Telephony Integration (CTI)?

Computer telephony integration (CTI) enables integration of third party media toolbars with Fusion Sales and Fusion Service. You can display a media toolbar by enabling the Partner CTI Service, and by giving your signed-in users the Access Partner Media Toolbar privilege which allows access to the toolbar.

Use CTI to:

- Make an outbound call using Click to Dial and skip manual dialing.
- Receive phone calls.

Integrating with CTI gives you 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

You can build this integration using the provided MCA Media Toolbar APIs or you can engage with one of Oracle's telephony partners. To connect with telephony partners, go to [Oracle Cloud Marketplace](#) and search for any one of the following:

- OpenMethods PopFlow
- OpenMethods Harmony (choose Harmony for B2B Service)
- Five9 (Cloud Contact Center for Oracle B2B Service)

How do I configure profile options to enable the Partner Media toolbar

The profile option `SVC_PARTNER_MEDIA_TOOLBAR_ENABLED` controls the visibility of the partner media toolbar. If you don't enable the profile option, the partner media toolbar is hidden. You can set this profile option either at a site level or at a user level.

To configure the profile options for enabling the partner media toolbar:

1. Navigate to the Setup and Maintenance work area and open the Tasks panel tab.
2. Click **Search** from the list of displayed tasks.
3. In the **Search** field, type `Manage Admin`.
4. From the displayed list of tasks, click **Manage Administrator Profile Values**.
5. In the Manage Administrator Profile Values page, navigate to the Search: Profile Option region.
6. In the Profile Option Code field, type `SVC_PARTNER_MEDIA_TOOLBAR_ENABLED`.
7. Click **Search**.

The Search Results: Profile Options region displays the list of profile options.

8. In the Profile Values region, specify the values in the **Profile Value** column for the **Site** Profile Level to 'Yes' if you want the partner media toolbar to be visible to all users.
9. Alternatively, set the Site level value to 'No' and add a new row for a user for whom the partner media toolbar must be visible.
10. Click **Save and Close**.

Note: Ensure that the user is assigned the privilege `Access Partner Media Toolbar` (`SVC_ACCESS_PARTNER_MEDIA_TOOLBAR_PRIV`) before trying to access the toolbar.

How do I configure Multichannel Architecture (MCA) profile options?

Profile options let you configure and control application data centrally. Use the Multichannel Architecture (MCA) profile options to manage the MCA runtime profile option values. Administrators and setup users manage MCA profile options in the Setup and Maintenance work area.

To configure the MCA Profile Options:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage MCA Profile Options
2. Click the name of the profile option you want to set.
This displays the **Manage MCA Profile Options** page for the selected profile option code.
3. In the Profile Values section for the profile option code, set the **Profile Value** for the Site **Profile Level**.
4. In the selected profile value, click the **New** icon to add a profile value.
5. Click **Save** or **Save and Close** to save the configuration.

MCA Profile Options

The following table lists the various MCA profile options that you can configure:

Profile Option	Description
SVC_MCA_DISABLE_JWT	Enables or disables the JSON Web Token generation.
SVC_MCA_DISABLE_OUTBOUND_CONTACT_VERIFY	Enables or Disables automatic display of contact verification screen for outbound calls. If enabled, it will override the call flow parameter configurations.
SVC_MCA_DISABLE_OUTBOUND_WRAP_UP	Enables or disables automatic display of the wrap-up screen for outbound calls. If enabled, it will override the call flow parameter configurations for wrap up. Note: For Sales for Redwood implementations, this setting must be enabled.
SVC_MCA_ENABLE_ENGAGEMENT_SYNC	Controls the in-focus chat interactions to be automatically associated with their screen popped tabs.
SVC_MCA_RECORD_ACTIVITIES_YN	Indicates whether activities must be logged for real time interactions. Note: For Sales for Redwood implementations, this setting must always be disabled (set to No).
SVC_MCA_RT_CHECK_EVENT_DISPATCH_INTERNAL_MS	Sets the number of milliseconds the user must wait for the start communication event. We recommend that you don't change this value unless instructed by Oracle Support.

Profile Option	Description
SVC_MCA_RT_EVENT_DISPATCH_MAX_RETRIES	Sets the number of times the user must wait for reverse lookup to return before running the start communication event. We recommend that you don't change this value unless instructed by Oracle Support.
SVC_MCA_SHOW_CONTACT_ON_WRAP_UP	Indicates whether to show the contact information during the wrap up process.
SVC_OVERRIDE_PARTNER_TOOLBAR_SELECTION	This option allows user to override system default setting of the partner media toolbar. User can select other toolbars from the list of active toolbars.
SVC_OVERRIDE_PARTNER_TOOLBAR_SELECTION	Enables the administrator to override the default setting of the partner media toolbar. The administrator can select other toolbars from the list of active toolbars.
SVC_PARTNER_MEDIA_TOOLBAR_ENABLED	Controls the visibility of the partner media toolbar. If it's not enabled, the partner media toolbar is hidden.
SVC_TOOLBAR_DEBUG_MODE_ENABLED	Turns on the JavaScript logging messages for debugging toolbar-related issues.

Import and Export Data in CSV Format

This topic describes how to import and export data from and into .CSV format files.

To import and export data::

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
2. Go to the relevant task, and click the **Actions** menu next to it. The tasks that you can perform are:
 - o Configure Call Flow Parameters
 - o Manage Application Classifications
 - o Manage Media Toolbar Configuration
 - o Manage Screen Pop Configuration
3. Click **Export Data > Create New** to export the data in a .CSV file format.
4. Click **Export Data > View All** to view all the exported .CSV files.
5. Click **Import Data > Create New** to import the data in a .CSV file format.
6. Click **Import Data > View All** to view all the imported .CSV files.

Related Topics

- [How do I configure screen pop pages?](#)
- [How do I configure the media toolbar?](#)
- [What's Computer Telephony Integration \(CTI\)?](#)
- [How do I update existing setup data?](#)

How do I configure the media toolbar?

The media toolbar is displayed in the following ways:

- 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.

The default toolbar gets assigned to all users.

Tip: Set the toolbar with the greater number of users as the default toolbar.

An administrator can also set other users to the non-default toolbar, or alternatively the agents can do this themselves.

An administrator can set the alternative toolbar for users by using the SVC_OVERRIDE_PARTNER_TOOLBAR_SELECTION profile option and creating a user row with the alternative toolbar for each user.

Agents can set an alternative toolbar by doing the following:

1. Click the **Settings and Actions** menu, and select **Set Preferences**.
2. Click **Media Toolbar**, and then click the **Override Media Toolbar** drop-down list and select the alternative toolbar.
3. Click **Save and Close**, then sign out and log back in.

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

Related Topics

- [What's Computer Telephony Integration \(CTI\)?](#)
- [How do I configure screen pop pages?](#)

Configure the Media Toolbar

To configure the media toolbar, do the following:

An administrator can configure the redwood toolbar. The media toolbar is displayed in the following ways:

- 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. The default toolbar gets assigned to all users.

Note: You can set the toolbar with the greatest number of users as your default toolbar.

An administrator can also set other users to the non-default toolbar, or alternatively the agents can do this themselves.

An administrator can set the alternative toolbar for users by using the `SVC_OVERRIDE_PARTNER_TOOLBAR_SELECTION` profile option and creating a user row with the alternative toolbar for each user.

Agents can set an alternative toolbar by doing the following:

1. Click the **Settings and Actions** menu, and select **Set Preferences**.
2. Click **Media Toolbar**, and then click the **Override Media Toolbar** drop-down list and select the alternative toolbar.
3. Click **Save and Close**, then log out and log back in.

Related Topics

- [Configure Screen Pop Pages](#)
- [What's Computer Telephony Integration \(CTI\)?](#)
- [Configure Business Objects](#)
- [How do I configure screen pop pages?](#)

CTI Media Toolbars APIs

Overview of the Media Toolbar API

Media toolbar APIs are a set of JavaScript functions delivered by Oracle to enable Computer Telephony Integration (CTI) partners to integrate their media toolbars with ServiceCenter.

These APIs facilitate communication between the media toolbar and ServiceCenter to exchange the following in response to inbound or outbound interaction requests:

- Configuration information
- Events
- Payload data

When enabled, ServiceCenter loads the toolbar and the media toolbar APIs in an iFrame within the browser.

How You Load the Media Toolbar API

Find toolbar API modules 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 the Oracle instance, 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*.
- **oraTbStyle**: Indicates whether the toolbar is displayed on a Live Window. **oraTbStyle=LiveWindow** indicates the Toolbar is Displayed in the Live Window, otherwise the value is set to Embedded to indicate the display is on the Applications page.

Here's an example to load the MCA Javascript dynamically:

```
<script type="text/javascript"
id="dynamicLoadScript">
(function() {
window.dynamicLoadCompleted = false;
window.staticLoadCompleted = false;
console.log('##DLS: Running DynamicLoadScript to load the Oracle API
JS...');
function getParameterByName(name) {
var match = RegExp('[?&]' + name +
'=([^\&]*)').exec(window.location.search);
return match && decodeURIComponent(match[1].replace(/\+/g, ' '));
}
var oraApiPath = '';
var oraOrigin = getParameterByName('oraParentFrame');
var oraApiSource = getParameterByName('oraApiSource');
console.log('##DLS: PageParameters - oraParentFrame: '+oraOrigin+'
oraApiSource: '+oraApiSource);
try {
if (oraApiSource && oraApiSource.startsWith("http")) {
oraApiPath = oraApiSource; //handle case where apiSource is loaded
from external CDN
} else {
oraApiPath = oraOrigin+oraApiSource;
}
console.log('##DLS: Oracle API JS URL: '+oraApiPath);
} catch (e) {
console.log('##DLS: Error '+e);
console.error(e);
}
function onDynamicLoadComplete() {
/* TODO INSERT ANY INIT PROCESSING HERE */
}
if (typeof define === 'function' && define.amd) {
try {
require([oraApiPath], function (mcaTlb) {
window['svcMca'] = {};
window.svcMca['tlb'] = mcaTlb;
window.dynamicLoadCompleted = true;
console.log('##DLS: Successfully loaded Oracle API JS');
onDynamicLoadComplete();
});
} catch (e) {
console.log('##DLS: Error loading ORACLE API JS from URL:
'+oraApiPath);
console.error(e);
}
} else {
var currentNode = document.getElementById('dynamicLoadScript');
var node = document.createElement('script');
node.type = 'text/javascript';
node.charset = 'utf-8';
node.async = true;
```



```
node.src = oraApiPath;
node.addEventListener('load', function(evt) {
window.dynamicLoadCompleted = true;
console.log('##DLS: Successfully loaded Oracle API JS');
onDynamicLoadComplete();
}, false);
node.addEventListener('error', function(evt) {
console.log('##DLS: Error loading ORACLE API JS from URL:
'+node.src);
}, false);
currentNode.parentNode.insertBefore(node, currentNode.nextSibling);
}) ();
</script>
```

Media Toolbar API Methods

Media toolbar APIs are divided into these methods:

- Configuration methods: Enable the toolbar and the application to exchange configuration information, and saves changes in application for retrieval by the toolbar when needed.
- Interaction methods: Provide notification on the events occurring on the toolbar, as well as exchange information used by the toolbar or Service Center.
- Toolbar windows management methods: Manage the toolbar windows and facilitate communication between windows.
- Event listeners: Allow the toolbar to register listeners for events triggered by Service Center.

The APIs provide a non-blocking interaction between the toolbar and Service Center. Each method has a callback function parameter that's invoked when a method on Service Center is complete, notifying the toolbar of the outcome.

What are the Media Toolbar APIs?

Use the media toolbar APIs with all toolbars that handle communication between agents and customers. The toolbar loads the APIs which are implemented in a JavaScript file format.

Note: 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. Use them to set up functionality for both the partner toolbar and ServiceCenter. There are three main flows of the toolbar interaction:

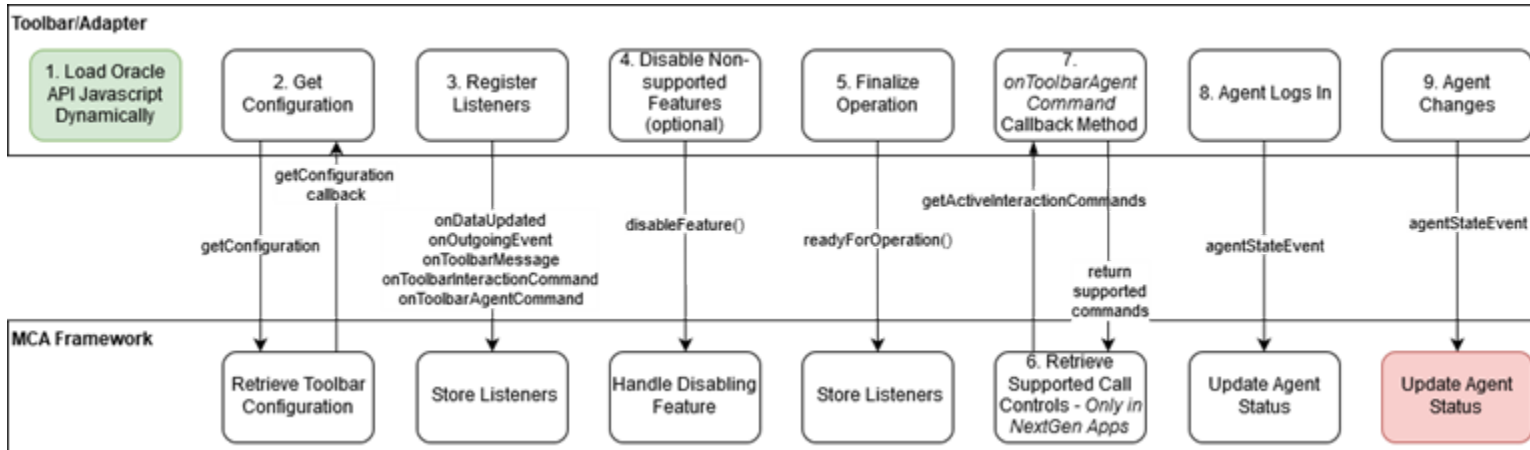
- Initial configuration of the partner toolbar and communication with Service Center.
- Inbound communication handling.
- Outbound communication handling.

The APIs are independent of communication channels, but use channel information as a parameter, passing it to Service Center to identify whether a toolbar supports multiple channels. You define and configure the channels using Functional Setup Manager. The APIs support both preconfigured and user-defined channels.

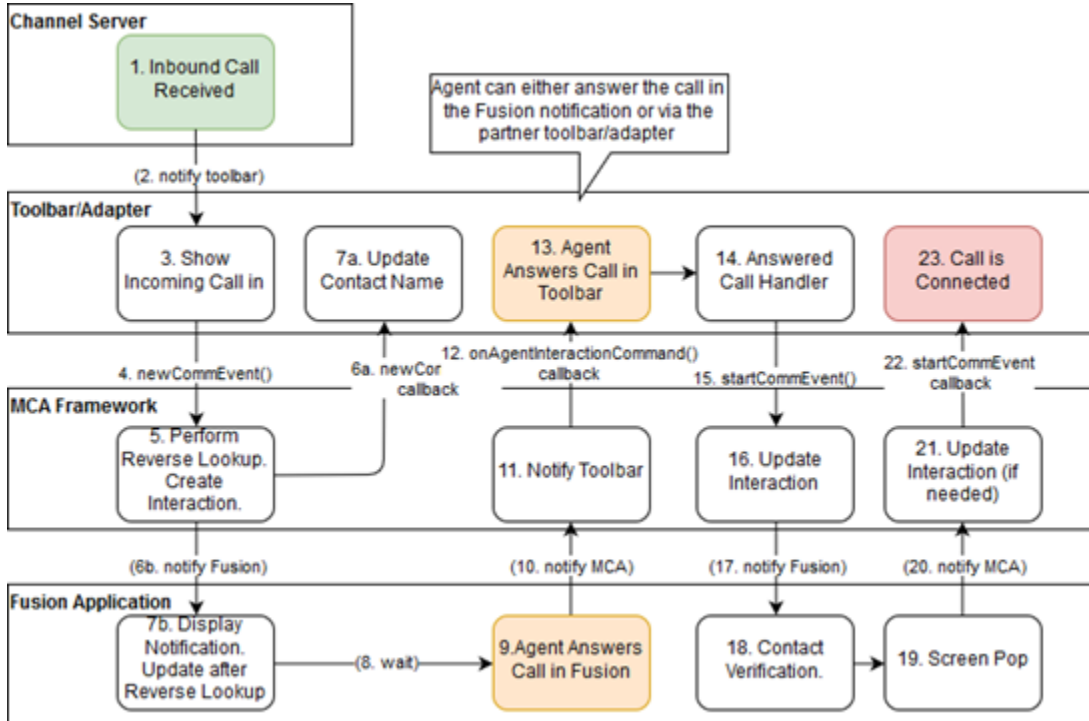
Data Flow Methods

The following figures show the flow of the methods that are called by the toolbar. The method calls can be required or optional. If a call is required, the toolbar implementation must make one call for each communication event, unless you specify otherwise. If a call is optional, the toolbar implementation can make a call whenever necessary, depending on the action needs.

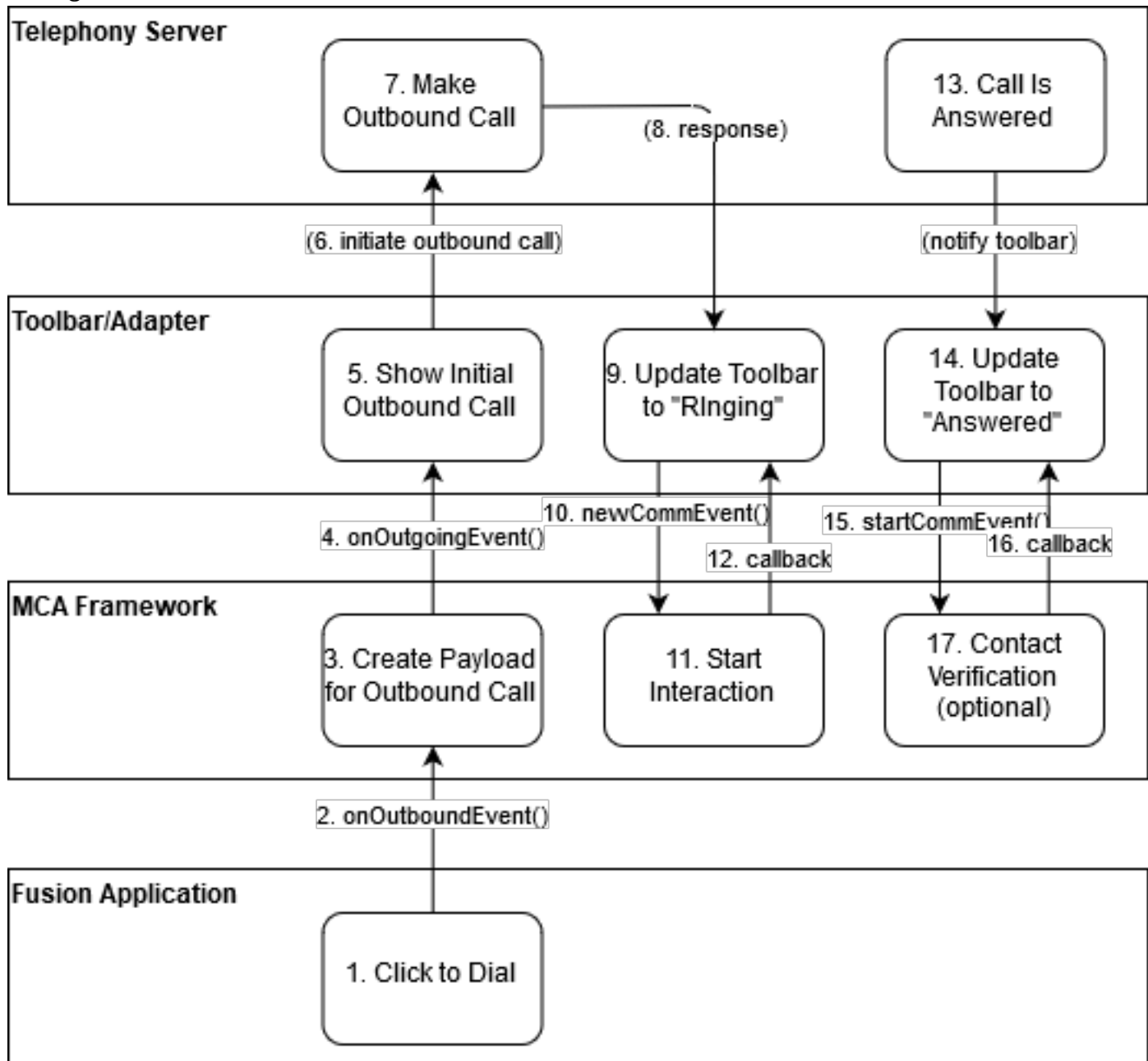
The following figure shows the configuration workflow.



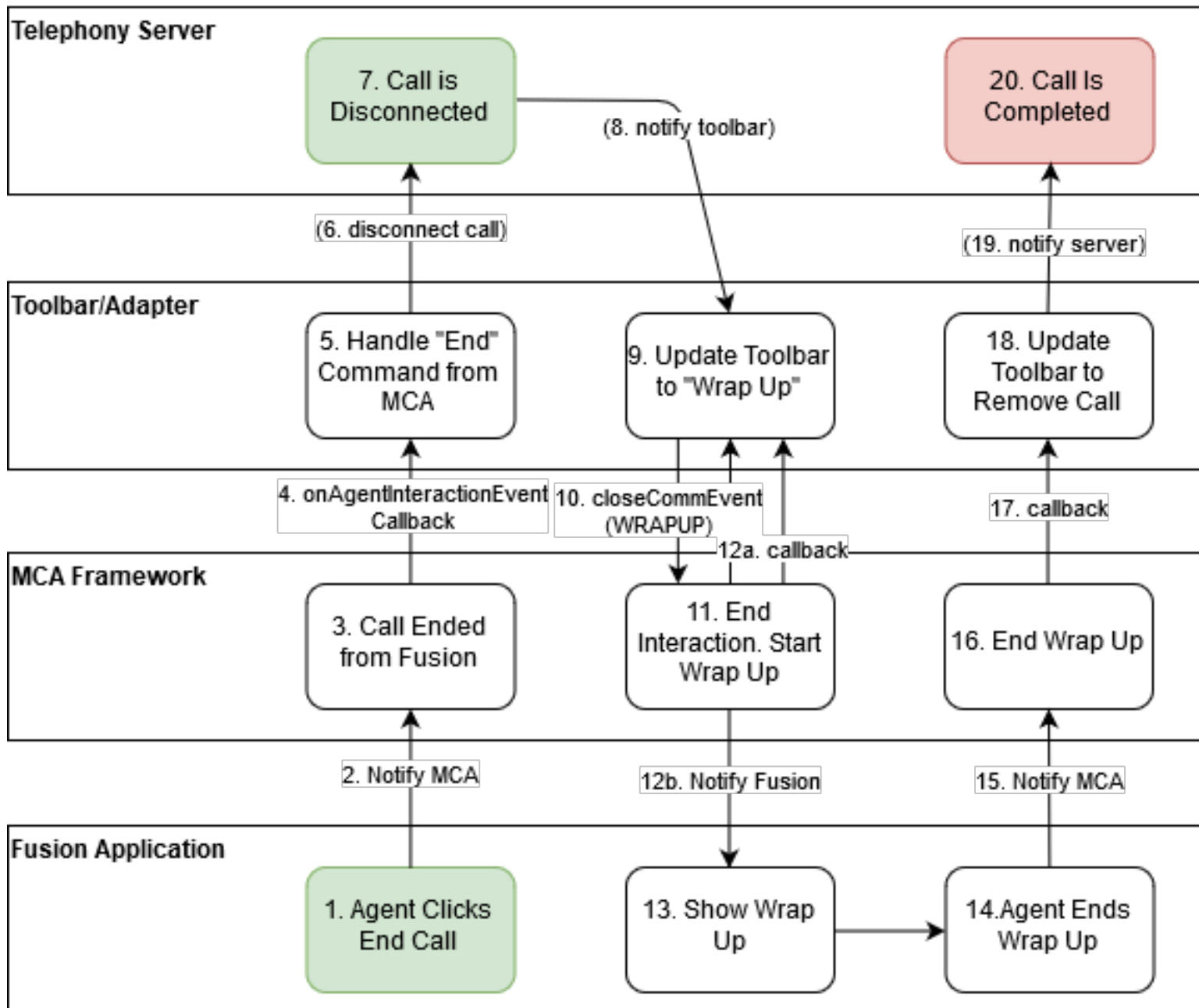
This figure shows the inbound call workflow.



This figure shows the outbound call workflow.



Call can be ended by agent (step 1), or by contact or adapter (step7)



Configuration APIs

Overview of Configuration APIs

The configuration APIs enable the toolbar and Service Center to exchange configuration data. They're listed in the following table.

Method	Usage
getConfiguration	Sets up the toolbar. The method retrieves a list of features that are supported by ServiceCenter.
disableFeature	Disables a feature that the toolbar doesn't support.
readyForOperation	Notifies ServiceCenter that the toolbar is ready for use, and that the agent is logged on to the toolbar, and ready to handle communication.

getConfiguration Method

Call this mandatory method to get configuration information that enables the toolbar to evaluate the features supported by ServiceCenter. Based on the evaluation, the toolbar informs ServiceCenter of any features that must be disabled. For more information, see the disableFeature API method.

Here's a list of the getConfiguration method parameters:

Parameter	Description
callback	Returns results to the caller toolbar, and retrieves configuration information in the form of a JSON formatted string.
configType	The type of configuration information returned by ServiceCenter. Possible values are TOOLBAR, TOKENS, PAGES, LOOKUP_OBJECTS, ALL, or FA_TOKEN. This is an optional parameter with a default value of TOOLBAR.

The return value is based on what the configType parameter passes. It includes the agent name along with the other information listed in the following table:

Parameter	Description
TOOLBAR	Returns a list of features supported by ServiceCenter. It also returns information about the toolbar, such as whether it's enabled, and if so, its dimensions. These values are preconfigured, but you can set up and modify them using Functional Setup Manager.
TOKENS	Returns information about both system and user-defined tokens. You can set up and modify these values using Functional Setup Manager.
PAGES	Returns information about both system and user-defined screen pop pages, including code, name, parameters, and associated tokens. You can set up and modify these values using Functional Setup Manager.
LOOKUP_OBJECTS	Returns information about business objects that are used to perform reverse lookup. Returns information about both system and user-defined screen pop pages, including code, name, parameters, and associated tokens. You can set up and modify these values using Functional Setup Manager.
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 ServiceCenter when calling the REST APIs. A specific call to this configType is required to obtain the JWT token.

Here's sample code to call the `getConfiguration` method

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

Here's a sample response for configuration data with configType `LOOKUP_OBJECTS`

```
{
  "lookupObjects": [
    {"name": "ServiceRequest"},
    {"name": "Account"},
    {"name": "Contact"}
  ]
}
```

Here's a sample response for configuration data with configType `FA_TOKEN`

```
{
```

```
" faTrustToken":  
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsIng1dCI6Im9ZcTdYODhqNXZwYWd3R2t2UGpMWjgxcloxdyJ9eyJleHAiOiE0MDg1NDAxMDIyNjEsImZU-i8vFx_XXx9E3iI915L_VD1VkXh96CLQPRYa5kW-ruLYeDOhhQL4M7ke8vPMArEp5rPoEOhJXnXYKKTJpCmE5RqJ_Zx00nPxJTY"  
}
```

disableFeature Method

This method informs ServiceCenter that a subset of available functionality must be disabled because the toolbar hasn't implemented it.

By default, the features sent by the getConfiguration API method are considered enabled. There's no return value with this method. The method parameters are described in the following table.

Parameter	Description
features	A string that contains the comma-separated feature names. For a list of features, see <i>API Features</i> .
callback	A method callback which returns results to the toolbar.

Here's sample code used to call the method:

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

readyForOperation Method

This method notifies ServiceCenter that the toolbar is ready for operation. A minimum of one call to this method is mandatory. The toolbar is disabled by default, and is enabled when this method is called with the readiness parameter set to TRUE.

Parameter	Description
readiness	Boolean value that specifies whether the toolbar is ready to operate.
callback	Callback which informs the toolbar that Service Center has received the readiness status message.

Here's sample code to call the `readyForOperation` method

```
<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>
```

Related Topics

- [API Features](#)

Interaction APIs

Overview of Interaction APIs

Interaction API facilitate communication between the toolbar and Service Center 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 a channel for which they make the call as a parameter. You configure the channels with Functional Setup Manager. For a list of provided channel names, see Channel Type Data. The following table lists the Interaction API methods.

Method	Usage
<code>newCommEvent</code>	Notifies the application that the toolbar has received a new <code>CommEvent</code> . This means that new communication has been assigned to or was initiated by the current agent. This call is mandatory.
<code>startCommEvent</code>	Notifies the application that the toolbar has started a <code>CommEvent</code> , where an agent has accepted the event.

Method	Usage
closeCommEvent	Notifies the application that the toolbar has closed the CommEvent. This call is mandatory.

newCommEvent Method

This method is called by the toolbar to inform ServiceCenter that a new communication event has started.

This is the first API call that the toolbar makes for a communication event. This method must provide through the eventId parameter, either a unique identifier or the media event identifier to be used for the communication event. For both inbound and outbound calls, it's important to specify the direction of the interaction. The newCommEvent must contain the following inData token to indicate the direction of the call: "SVCMA_COMMUNICATION_DIRECTION". Valid values are: "ORA_SVC_INBOUND" and "ORA_SVC_OUTBOUND". This method call is mandatory.

For inbound calls, the primary purpose of the call is to identify customers by using reverse lookup functionality and to trigger the incoming call notification. The toolbar uses tokens to pass all the information from the Interactive Voice Response systems. All customer information retrieved is sent back to the toolbar using system tokens.

Here's a list of the newCommEvent parameters.

Parameter	Description
channel	Name of the channel for which the method is called. For more information on what channels are available, see <i>Channel Type Data</i> .
appClassification	Name of the application classification defined for the current toolbar. If this parameter is passed as null, it's assumed that there's no application classification. For a list of preconfigured values, see <i>Application Classification Code</i> .
eventId	A toolbar-generated media event identifier. The same parameter value must be used for all method invocations for a communication. This parameter is stored with the internal interaction record.
inData	An object containing information about incoming events for customer identifiable data, along with attributes containing the names of system or user-defined defined tokens. To specify a timeout value for an incoming call notification, add the following pair to this parameter: SVCMA_OFFER_TIMEOUT_SEC, #OFSECOND Replace the #OFSECONDS value with the actual number of seconds where you want to start the timer. When the timer expires the onDataUpdated() method is invoked with the updateType parameter set to OFFER_TIMEOUT. The toolbar then invokes the closeCommEvent() method to end the interaction.
callback	Returns results to the caller toolbar.
lookupObject	A business object used to run the reverse lookup. Use this optional parameter to get customer information. The default value of Contact. For more information about the list of system lookup objects, see <i>System Business Objects</i> .

Parameter	Description
channelType	The type of channel for which the method is being called. Note that if, for example, the channel is PHONE but the channelType isn't provided, the default value is set to ORA_SVC_PHONE. For more information, see <i>Channel Type Data</i> .

Here's a list of member attributes of the response object. The object has no return value.

Member Attribute	Description
result	The server side running status. Result values are success or error.
outData	Represents an object containing possible identified data from Service Center for the contact or for the organization. This attribute contains the names of system tokens.
error	Contains error message if the result value is an error. If not, then the value is undefined. For more information about error message codes, see Error Messages.

Note: The newCommEvent method triggers the creation of an interaction object. This object contains information related to communication, and objects that are related to communication. The identifier for the interaction is passed back with the callback response as the value for the SVC_MCA_INTERACTION_ID token. Subsequent invocations of the method must send this token and its value, along with other IVR data to Service Center.

When the newCommEvent method is invoked for a transfer, the toolbar uses the SVC_MCA_PARENT_INTERACTION_ID token to send IVR data and the originating communication interaction ID to Service Center which associates the communication events. In addition, you must also specify the SVC_MCA_2NDARY_CALL_TYPE token to indicate whether this call is a Transfer, a Consultation, or a Conference.

Here's an example of code used to call this method.

```
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
<script type="text/javascript">
function inboundCallNotification() {
  var inData = {};
  inData.SVC_MCA_ANI = '5551234';
  inData.SVC_MCA_CONTACT_ID = '1234567890';
  svcMca.tlb.api.newCommEvent('PHONE', 'ORA_SERVICE', '12345-1234-67890', inData, null, function (response) {
    if (response.result == 'success') {
      console.log('Customer: ' + response.outData.SVC_MCA_CONTACT_NAME + ' (' + response.outData.SVC_MCA_CONTACT_ID
        + ')');
      console.log('Account: ' + response.outData.SVC_MCA_ORG_NAME + ' (' + response.outData.SVC_MCA_ORG_ID + ')');
      alert('Success! Results available in log.');
```

</html>

startCommEvent Method

The toolbar makes this mandatory method call when an agent accepts a new communication that was triggered by the `newCommEvent()` call.

This method invokes the Identify Contact flow, then triggers the Customer Validation flow, and then runs the screen pop which the following explains:

1. The method first triggers the Customer Validation flow.

This 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 `SVC_MCA_BYPASS_IDENTIFY_CONTACT`. Toolbars can then use the `getCustomerData()` method to manually request contact identification.

2. The method then triggers the Customer Verification flow, an optional flow that opens the customer verification page. If the agent doesn't verify the caller, no screen pop is performed. Customer verification can be bypassed by passing in the `SVC_MCA_BYPASS_CUSTOMER_VERIFICATION` token in the call data.
3. Finally, the method runs a screen pop using your predefined rules. If you want the toolbar to prevent the screen pop, you can add the `SVC_MCA_BYPASS_AUTO_SCREENPOP` token into the call data, and use the `invokeScreenPop()` method to manually trigger the pop.

Note: A callback passed with the `startCommEvent` method can be invoked multiple times if the Identify Contact and Customer Verification flows triggered. The `outData` information from the response can contain updated customer information.

Here's the method details:

Parameter	Description
channel	The name of the channel for which the method is called. For more information on what channels are available, see <i>Channel Type Data</i> .
appClassification	The name of the application classification defined for the current toolbar. If this parameter is passed as null, it's assumed that there is no application classification. For a list of preconfigured values, see <i>Application Classification Code</i> .
eventId	The toolbar 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	An object containing information about incoming events for customer identifiable data, as well as attributes containing the names of system or user-defined defined tokens.
callback	Returns results to the caller toolbar.

Parameter	Description
channelType	The type of channel for which the method is being called. Note that if, for example, the channel is PHONE but the channelType isn't provided, the default value is set to ORA_SVC_PHONE. For more information, see <i>Channel Type Data</i> .

Here's the member attributes of the response object. The object has no return value.

Member Attribute	Description
result	The server side running status. Result values are success or error.
outData	Represents an object containing possible identified data from Service Center for the contact or for the organization. This attribute contains the names of system tokens.
error	Contains the error message if the result value is an error. If not, then the value is undefined. For more information about error message codes, see Error Messages.

```
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
<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) {
    if (response.result == 'success') {
      console.log('Created service request: '+response.SVCMCA_SR_NUMBER);
      alert('Success! Results available in log.');
```

closeCommEvent

The toolbar calls this method when communication ends. The parameter reason is used to indicate how the communication ended and provide more information. Value reason values are shown in the following list:

- ABANDONED: Indicates that the customer ended the communication before the agent accepted and before timeout.

- **CANCELED:** Mark outbound call for which the communication started, but subsequently was canceled by the agent before it was answered by the customer.
- **ENDCOMMUNICATION:** Indicates that the communication ended and if Service Center is in wrap-up mode it then must close the wrap-up page.
- **MISSED:** The communication attempt couldn't be established for another reason.
- **REJECT:** When an agent rejects a communication.
- **TIMEOUT:** The communication attempt timed out.
- **WRAPUP:** Displays the Wrap Up window to indicate to the agent that the communication must come to a close.

Here's the parameters for this method:

Parameter	Description
channel	The name of the channel for which the method is called. For more information on what channels are available, see <i>Channel Type Data</i> .
appClassification	The name of the application classification defined for the current toolbar. If this parameter is passed as null, it's assumed that there's no application classification. For a list of preconfigured values, see <i>Application Classification Code</i> .
eventId	Toolbar 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	An object containing information about incoming events for customer identifiable data, along with attributes containing the names of system or user-defined defined tokens.
reason	ABANDONED, CANCELED, ENDCOMMUNICATION, MISSED, REJECT, TIMEOUT, OR WRAPUP.
callback	Returns results to the caller toolbar.
channelType	The type of channel for which the method is being called. Note that if, for example, the channel is PHONE but the channelType isn't provided, the default value is set to ORA_SVC_PHONE. For more information, see <i>Channel Type Data</i> .

Here's the member attributes of the response object. The object has no return value.

Member Attribute	Description
result	The server side running status. Result values are success or error.
outData	Represents an object containing possible identified data from Service Center for the contact or for the organization. This attribute contains the names of system tokens.
error	Contains the error message if the result value is an error. If not, then the value is undefined. For more information about error message codes, see Error Messages.

Here's sample code for this method:

```
<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.');
```

Event Listener APIs

Overview of Event Listener APIs

The event listener API allows the toolbar register listeners to track events that are triggered by Service Center. This allows Service Center to initiate outbound interaction events and send updates to the toolbar.

Here's a list of the Event Listener APIs and their usage.

Method	Usage
agentStateEvent	Notifies the application of changes in a user's signed in or availability status for the specified channel.
onDataUpdated	Registers a callback for customer data update events to be transmitted to the media toolbar with the updated information.
focusInteractionEvent	Notifies the application that the toolbar has changed the previously active interaction slot to a different interaction slot.
onOutgoingEvent	Registers a toolbar-initiated callback for the start of an outgoing event triggered from Service Center.
outboundCommError	Notifies Service Center that an error occurred during initiation of the outbound event.
onToolbarAgentCommand	Registers a listener with Service Center to provide agent control functionality.

Method	Usage
onToolbarInteractionCommand	Registers a listener with Service Center to provide interaction control functionality.

agentStateEvent Method

This method notifies Service Center of a change in the user's signed in or availability status for the specified channel.

Note: In order to properly reflect the agent state on the phone icon at the top of the application window, it's important to inform Service Center of the agent's current state: logged in, logged out, available, or on break. Upon establishing the initial connection to the toolbar, be sure to call `agentStateEvent()` passing the correct values for `isLoggedIn` and `isAvailable`. All state changes must be conveyed using the same method.

Here's a list of the method parameters:

Parameter	Description
channel	Name of the channel. (Required.)
eventId	Unique identifier for the event. (Required.)
isAvailable	Boolean to indicate if the user is available to receive routed assignments for the specified channel. (Required.)
isLoggedIn	Boolean to indicate if the user is signed in to the specified channel. (Required.)
stateCd	System code to indicate the current availability status of the user. (Optional.) For future use.
stateDisplayString	Translatable string to indicate the user's current channel activity state in the Service Center UI, such as idle, busy, on break on so on. This must be translated to the user's language. (Required.)
reasonCd	System code to indicate the reason for the current availability status. (Optional.) For future use.
reasonDisplayString	Translatable string to display the reason for the current status on the Service Center UI. (Optional.) For future use.
inData	Object containing the name value pairs if additional details are required.
callback	Function on the toolbar to indicate event receipt by Service Center Service.
channelType	The type of channel. (Required.)

The following is sample code to call this method.

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

onDataUpdated Method

The toolbar calls this method to register the callback listener for a data update which occurred in Service Center which must be transmitted to the toolbar.

Using this listener, Service Center can send notifications about updates of Service Center contact information which will then be reflected in the toolbar. Additionally, Service Center can use this method to notify the toolbar that Service Center actions, such as a Wrap-up, have been completed.

Here's the parameters for this method:

Parameter	Description
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, it's assumed that there are no application classifications. For a list of ready to use values, see <i>Application Classification Code</i> .
callback	Method callback to inform the caller toolbar of the results.
channelType	The type of channel for which the method is being called. For instance, if the channel is PHONE and the channelType isn't provided, the default value is set to ORA_SVC_PHONE. For more information, see <i>Channel Type Data</i> .

The response object has the following member attributes.

Attribute	Description
updateType	Type of update. The values can be <code>CUSTOMER_DATA</code> or <code>WRAPUP_CLOSED</code> or <code>OFFER_TIMEOUT</code> .
eventId	Media event identifier that's used to identify which communication event is targeted by this update.
outData	Object that has attributes with the names of system token, such as <code>SVCMCA_CONTACT_ID</code> , <code>SVCMCA_CONTACT_NAME</code> , <code>SVCMCA_ORG_ID</code> , and <code>SVCMCA_ORG_NAME</code> .

Here's sample code to call the method:

```
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
<script type="text/javascript">
function onDataUpdated() {
  svcMca.tlb.api.onDataUpdated('PHONE','ORA_SERVICE', function (response) {
    console.log('Media Event: ' +response.eventId);
    console.log('Contact number: ' +response.outData.SVCMCA_CONTACT_ID);
    console.log('Contact name: ' +response.outData.SVCMCA_CONTACT_NAME);
    console.log('Organization number: ' +response.outData.SVCMCA_ORG_ID);
    console.log('Organization name: ' +response.outData.SVCMCA_ORG_NAME);
    alert('Customer data got updated, see in the log. ');
  }, 'ORA_SVC_PHONE');
}</script>
</head>
<body>
<input type="button" value="Register data update listener" onclick="onDataUpdated()" />
</body>
</html>
```

onOutgoingEvent Method

The toolbar calls this method to register the callback listener for Service Center which triggers an outgoing communication event. Using this callback Service Center passes the information required for the toolbar to perform the outgoing event.

Here's the parameters for this method:

Parameter	Description
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, it's assumed that there are no application classifications. For a list of ready to use values, see <i>Application Classification Code</i> .
callback	Function used by Service Center on the toolbar which indicates receipt of an Service Center event.

Parameter	Description
channelType	The type of channel for which the method is being called. For instance, if the channel is PHONE and the channelType isn't provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data.

The response passed in the callback has attributes with system token names, such as `SVC_MCA_ANI`, `SVC_MCA_EMAIL`, `SVC_MCA_CONTACT_ID`, and `SVC_MCA_CONTACT_NAME`. There are no returns.

Here's sample code to call the method:

```
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
<script type="text/javascript">
function onOutgoingEvent() {
  svcMca.tlb.api.onOutgoingEvent('PHONE','ORA_SERVICE', function (response) {
    console.log('Calling: '+response.SVC_MCA_ANI);
    console.log('Contact name: '+response.SVC_MCA_CUSTOMER_NAME);
    alert('Initiating outgoing call, see information in the log.');
```

```
  }, 'ORA_SVC_PHONE');
}d>
</body>
<input type="button" value="Register outcall listener" onclick="onOutgoingEvent()" />
</body>
</html>
```

outboundCommError Method

The toolbar calls this method to notify Service Center that an error occurred during initiation of the outbound event. The error occurs if the identifier of the event, such as phone number, or email can't be used to establish a connection.

Here's the parameters for this method:

Parameter	Description
channel	Name of the channel for which the method is called.
commUuid	A unique identifier sent from Service Center to the toolbar containing the outgoing event notification.
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	Error message displayed to the user.
callback	Function used by Service Center on the toolbar which indicates receipt of an Service Center event.

Parameter	Description
channelType	The type of channel for which the method is being called. For instance, if the channel is PHONE and the channelType isn't provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data.

The response object has the following member attributes. There are no returns.

Attribute	Description
result	Represents the status of the method on the server side. Values are success and error.
error	Contains error messages. If there's no error, the value is undefined. For more information about error message codes, see Error Messages.

Here's sample code to call the method:

```
<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 Method

This method registers a listener with Service to give the agent control functionality. The method is only called once during initialization.

Here's the parameters for this method:

Parameter	Description
channel	Name of the channel for which the method is called.
channelType	The type of channel for which the method is being called. For instance, if the channel is PHONE and the channelType isn't provided, the default value is set to ORA_SVC_PHONE. For more information, see Channel Type Data.

Parameter	Description
executor	The function implemented by the toolbar which is used by the <i>agentCommandPrototype</i> object.

The *agentCommandPrototype* object has these attributes:

Attribute	Description
eventId	Represents the status of the method on the server side. Values are success and error.
command	Contains error messages. If there is no error, the value is undefined. For more information about error message codes, see Error Messages. The following commands are available: • <code>getCurrentAgentState</code> • <code>getActiveEngagements</code>: The <code>outData</code> must contain the <code>activeCount</code> number and the <code>engagements</code> array value. For example: <code>{ activeCount:1, engagements:[{eventId:"1234"}] }</code>. If there are no active engagements, the <code>engagements</code> parameter must be an empty array. • <code>makeAvailable</code> • <code>makeUnavailable</code>
channel	The channel name.
channelType	The type of channel.
inData	The object containing the name value pairs for the command parameters.
result	Populated by the toolbar after the command process has completed successfully. If there was a failure, the reason will be displayed.
resultDisplayString	Error displayed to the user, and populated by the toolbar.
outData	Populated by the toolbar if the command required output data.
sendResponse	Callback function used upon completion of the command processing. The <code>sendResponse</code> method must be passed to the command object with the result populated. Also the <code>outData</code> and <code>resultDisplayString</code> attributes must be populated if required.

Here's sample code to call the method:

```
<html>
<head>
<script type="text/javascript" src="http://domain:port/ora_support/js/mcaInteractionV1.js">
</script>
<script type="text/javascript">
function agentCommandExecutor(command) {
var cmd = command.command;
switch(cmd) {
case "getCurrentAgentState":
```

```
command.outData = {
  '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>
```

onToolbarInteractionCommand Method

Registers a listener with Service Center to provide interaction control functionality. This must be called once during initialization. The executor function is used with an `interactionCommandPrototype` object.

The *interactionCommandPrototype* object has the following attributes:

Attribute	Description
eventId	Unique media item identifier supplied by the toolbar during the <i>newCommEvent</i> .
command	Name of the command to execute. The following commands are available: - accept: Accepts the incoming engagement. - reject: Rejects the incoming engagement. - setActive: Discloses the specified engagement slot to the toolbar.
slot	An optional toolbar identifier supplied by the toolbar during the <i>newCommEvent</i> .
inData	The object containing the name value pairs for the command parameters.

Attribute	Description
result	Populated by the toolbar after the command process has completed successfully. If there was a failure, the reason will be displayed.
resultDisplayString	Error displayed to the user, and populated by the toolbar if the command
outdata	Populated by the toolbar if the command required output data.
sendResponse	Callback function used upon completion of the command processing. The sendResponse method must be passed to the command object with the result populated. Also the outData and resultDisplayString attributes must be populated if required.

The following is sample code to call the method:

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

function interactionCommandExecutor(command) {
switch(command.command)
case "accept":
alert("Received accept command");
break;
case "reject":
alert("Received reject command");
break;
case "setActive":
alert("Received setActive command");
break;
}
command.result = 'success';
command.sendResponse(command);
}

function registerInteractionCommandListener() {
svcMcaTlb.api.onToolbarInteractionCommand(interactionCommandExecutor);
}
</script>
<body>
<input type="button" value="Register interaction command listener"
onclick="registerInteractionCommandListener()" />
</body>
</html>
```

API Features

Features are functionality or sets of functionality that are supported by Service Center, and can be implemented by a toolbar. Features are supported by the API and underlying Service Center implementation. The following table lists the features of the API:

Feature Code	What you can do with the Feature
INBOUND_CALL	Handle messages related to inbound calls.
OUTBOUND_CALL	Handle messages related to outbound calls. The API notifies the toolbar to initiate the call using the supplied information.
TRANSFER_CALL	Handle messages related to a call transfer.
CONFERENCE_CALL	Handle messages related to a call conference.
OUTBOUND_CALL_EXTENSIONS	Handle messages related to outbound calls. The API notifies the toolbar to initiate the call with the specified phone number and extension.

Configure Screen Pop Pages

Configure Business Objects

You can specify the business objects for reverse lookup and screen pop using business object configuration. This includes configuring a standard business object or a user business object.

A standard object is based on an out-of-the-box business object. A user business object is based on a user-defined business object that's created using the Application Composer. For more information about creating user business objects using Application Composer, see the Oracle Applications Cloud Configuring Applications Using Application Composer guide.

To configure a business object:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Screen Pop Configuration

The Screen Pop Configuration page is displayed.

2. Select the tab, Business Objects. The Manage Business Objects Configuration page is displayed.
3. View the business objects that are read only, and user objects that can be created, modified, and deleted.

Create a Standard User Business Object

A standard business object is based on a previously configured 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 previously configured 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**.

Create 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 Applications Cloud Configuring Applications Using Application Composer 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**.

Related Topics

- [How do I configure the media toolbar?](#)
- [What's Computer Telephony Integration \(CTI\)?](#)
- [How do I configure screen pop pages?](#)
- [Configure Screen Pop Pages](#)
- [How do I update existing setup data?](#)

How do I configure screen pop pages?

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.

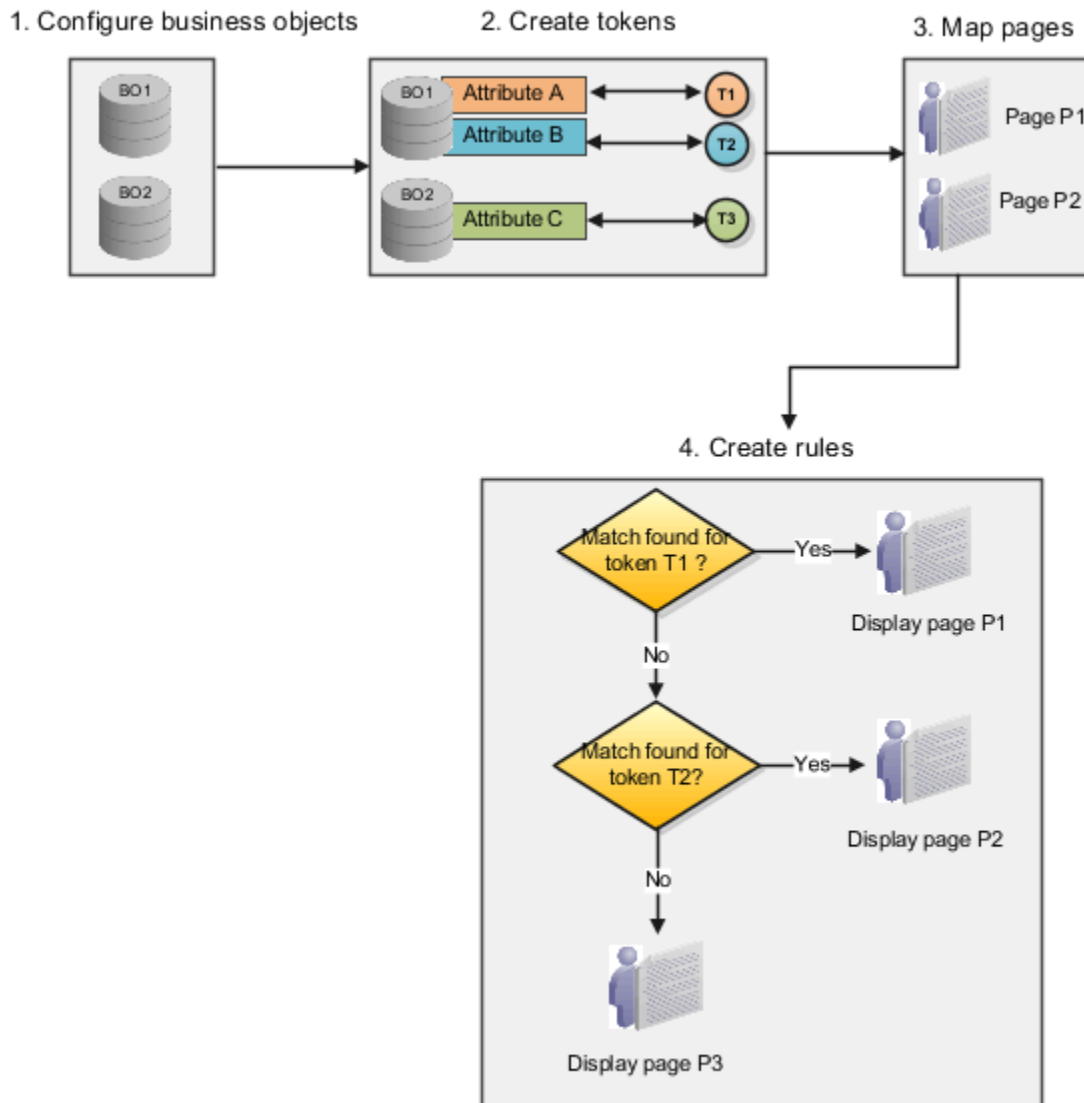
You can create screen pop pages for ready-to-use standard business objects or for user-defined objects. To create screen pop for user-defined objects, you must first create the objects, define the fields, pages, and other elements using Application Composer.

You configure screen pop pages in Setup and Maintenance using the following steps:

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 identify the page that's 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's 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:



Configure Screen Pop Pages

This topic describes how to configure screen pop pages.

Configuring a screen pop page includes the following steps:

1. Configuring a business object
2. Creating a token
3. Mapping a page
4. Defining rules

To configure a screen pop page:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Screen Pop Configuration

Note: You may have to select All Tasks from the Show drop down list to view this task.

The Screen Pop Configuration page is displayed.

Create a Business Object

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 a ready-to-use business object and a user business object is based on a user-defined business object that's created using the Application Composer. For more information about creating user business objects using Application Composer, see *Configuring Applications Using Application Composer* guide.

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

To view a standard business object:

1. Click the **Business Objects** tab.
Business Objects tab lists standard and user-defined business objects. The standard or the predefined objects cannot be modified or deleted.
2. Click the **Systems Objects** tab.
3. View the mapped system objects.
4. Click **Save** or **Save and Close**.

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 *Configuring Applications Using Application Composer* guide.

To create a user business object:

1. Click the **Business Objects** tab.
Business Objects tab lists standard and user-defined business objects. The standard or the predefined objects cannot be modified or deleted.
2. Click the **User Business Objects** tab.
3. Click the add icon.
4. Select object type as **User-Defined**.
5. Select the application in which you created an object. For example, select **Service** application.
6. 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.

7. Click **Add**.
8. Click **Save** or **Save and Close**.

Create 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's 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's associated with the token. Attribute is an optional value.
4. Specify the values for the token and click **Save**.

Map Screen Pop Page

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

1. On the **Screen Pop Page Configuration** page, click the **Pages** tab.
2. To create a page mapping, click the add icon.
3. Click the **User Interface Type** drop down list and select **Application Development Framework**.
4. Select a **Standard** or a **User-Defined** type.
5. Select an **Object Name**.
6. Enter a **Page Name**.
7. 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.
8. 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.
9. In the Page Parameters Mapping section, associate required parameter with a token or a user-defined value.
Based on the input that's 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.
10. Click **Save**.

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

Define 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 isn't 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. Click the **User Interface Type** drop down list, and select either **Application Development Framework** or **Redwood** depending on your implementation.
 - e. A rule set is **Active** by default. To deactivate a rule set, clear the **Active** option.
 - f. 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**.

Related Topics

- [How do I configure screen pop pages?](#)
- [How do I configure the media toolbar?](#)
- [What's Computer Telephony Integration \(CTI\)?](#)
- [How do I update existing setup data?](#)

Create Lookup Filters

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.

When customers make service calls, they provide information such as Customer ID, Date of Birth, and First Name through an Interactive Voice Response (IVR) tool. Customer details are extracted from the database based on the availability of the token value, starting from the token of highest priority.

Use this page to configure reverse lookups and lookup filters for channel driven interactions with customers. Order the filters by the priority in which the tokens are evaluated. To create a new lookup filter, you can duplicate an existing one and make modifications. To create a new lookup filter from scratch, perform the following steps:

1. On the **Screen Pop Page Configuration** page, click the **Lookup Filters** tab.
2. Click the Add icon to create a lookup set.
 - a. Enter a **name**.
 - b. Select an **Application Classification** from the drop-down list.
 - c. Select the **Active** option to activate a lookup set.
 - d. Enter a **Description**.
3. 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 or select **Move Up** or **Move Down** option.
4. Click **Save** or **Save and Close**.

System Tokens

System tokens include the following tokens types.

- Data tokens: Represent data values in the Multichannel Application
- Control tokens: Control the flow of an interaction

The following table lists the data tokens.

Data Token Name	Token Code	Description
Asset Id	SVCMCA_ASSET_ID	Asset Identifier.
Asset Name	SVCMCA_ASSET_NAME	Asset Name.
Asset Number	SVCMCA_ASSET_NUM	Asset Number.
Channel Id	SVCMCA_CHANNEL_ID	Channel Identifier which is populated from Channel Name to be passed for page parameters.
Communication direction	SVCMCA_COMMUNICATION_DIRECTION	Communication direction, which can either be inbound or outbound. Set the value to ORA_SVC_INBOUND for inbound and ORA_SVC_OUTBOUND for outbound direction.

Data Token Name	Token Code	Description
		These values are defined in the ORA_SVC_IM_DIRECTION_CD Lookup Type.
Communication Reason	SVCMCA_WRAPUP_COMM_REASON	Communication Reason of the Interaction
Contact First Name	SVCMCA_CONTACT_FIRST_NAME	Contact's first name.
Contact ID	SVCMCA_CONTACT_ID	Contact person identifier. This ID value might be available on the customer documentation or it can be retrieved from the application for use.
Contact Job Title	SVCMCA_CONTACT_JOB_TITLE	Contact's job title.
Contact Last Name	SVCMCA_CONTACT_LAST_NAME	Contact's last name.
Contact Name	SVCMCA_CONTACT_NAME	Contact name. Obtained from Oracle for use.
Contact Organization ID	SVCMCA_CONTACT_ORG_ID	Contact's primary organization ID.
Contact Organization Name	SVCMCA_CONTACT_PRIM_ORG_NAME	Contact's primary organization name.
Contact Phone Number	SVCMCA_CONTACT_PRIMARY_PHONE	Contact's primary phone number.
Display Name	SVCMCA_DISPLAY_NAME	Contact name or other identifying contact information to be displayed in supplier toolbars.
Email	SVCMCA_EMAIL	Contact email. Can be obtained from IVR or from Oracle for use.
Interaction identifier	SVCMCA_INTERACTION_ID	Interaction identifier. Obtained from the application for use. Can be sent from the application using the APIs to prevent creating duplicate interactions.
Interaction Notes	SVCMCA_WRAPUP_INTERACTION_NOTES	Notes of the Interaction
Lead ID	SVCMCA_LEAD_ID	Lead Id. Can be obtained from IVR system or from FA module and reused later.
Lead Name	SVCMCA_LEAD_NAME	Lead name. Usually, will be obtained from FA module and reused later.
Loyalty Member ID	SVCMCA_LOYALTY_MEMBER_ID	Loyalty Member Identifier.
Loyalty Member Name	SVCMCA_LOYALTY_MEMBER_NAME	Loyalty Member Name.

Data Token Name	Token Code	Description
Loyalty Member Number	SVCMCA_LOYALTY_MEMBER_NUMBER	Loyalty Member Number.
Mobile Phone Area Code	SVCMCA_MOBILE_AREA	Mobile Phone Area Code. Token value from the call data that can be passed to the contact picker when creating a contact.
Mobile Phone Country Code	SVCMCA_MOBILE_CC	Mobile Phone Country Code. Token value from the call data that can be passed to the contact picker when creating a contact.
Mobile Phone Local Number	SVCMCA_MOBILE_LOCAL	Mobile Phone Local Number. Token value from the call data that can be passed to the contact picker when creating a contact.
Object ID List	SVCMCA_OBJ_ID_LIST	Object Id List for filtered display of specific objects.
Opportunity ID	SVCMCA_OPPORTUNITY_ID	Opportunity Id. Can be obtained from IVR system or from FA module and reused later.
Opportunity Name	SVCMCA_OPPORTUNITY_NAME	Opportunity name. Usually, will be obtained from FA module and reused later.
Organization ID	SVCMCA_ORG_ID	Account or organization ID. The ID may be available on the customer documentation. The ID can be identified by Interactive Voice Response (IVR) or can be retrieved from Oracle for use.
Organization Name	SVCMCA_ORG_NAME	Account name. Obtained from Oracle for use.
Phone Area Code	SVCMCA_PHONE_AREA	Work Phone Area Code. Token value from the call data that can be passed to the contact picker when creating a contact.
Phone Country Code	SVCMCA_PHONE_CC	Work Phone Country Code. Token value from the call data that can be passed to the contact picker when creating a contact.
Phone Extension	SVCMCA_PHONE_EXT	Work Phone Extension. Token value from the call data that can be passed to the contact picker when creating a contact.
Phone Local Number	SVCMCA_PHONE_LOCAL	Work Phone Local Number. Token value from the call data that can be passed to the contact picker when creating a contact.

Data Token Name	Token Code	Description
Phone Number	SVCMCA_ANI	Automatic Number Identification value sent from telephony provider.
Resolution Code	SVCMCA_WRAPUP_RESOLUTION_CODE	Resolution Code of the Interaction.
Service Request Identifier	SVCMCA_SR_ID	Service request identifier. Obtained from Oracle for use.
Service request Number	SVCMCA_SR_NUM	Service request number to be passed whenever the number is available.
Service Request Title	SVCMCA_SR_TITLE	Service request title. Obtained from the application for use.
Task ID	SVCMCA_TASK_ID	Task ID.
Transfer Reason	SVCMCA_WRAPUP_TRANSFER_REASON	Transfer Reason of the Interaction
Universal Work Object Identifier	SVCMCA_UWO_ID	Specifies the universal work object identifier.
Work Order Id	SVCMCA_WORK_ORDER_ID	Work Order Identifier.
Work Order Number	SVCMCA_WORK_ORDER_NUM	Work Order Number.

The following table lists the data tokens for HR HelpDesk use.

Data Token Name	Token Code	Description
Person Email	SVCMCA_PER_EMAIL	Person Email address
Person Id	SVCMCA_PER_PERSON_ID	Person Unique Identifier
Person List Name	SVCMCA_PER_LIST_NAME	Person last name, first name
Person Name	SVCMCA_PER_NAME	Person display name
Person National Id Number	SVCMCA_PER_NAT_ID_NUM	Person's National Id Number
Person Number	SVCMCA_PER_NUMBER	Person Number
Person Phone	SVCMCA_PER_PHONE	Person Phone number

The following table lists the control tokens.

Control Token Name	Token Code	Description
Parent interaction identifier	SVMCA_PARENT_INTERACTION_ID	Parent interaction identifier in the case of a transfer. Expected to be sent along a new communication event data of the second call.
Secondary Call Type	SVMCA_2NDARY_CALL_TYPE	A type of secondary party call, which can be one of Transfer, Conference, and Consult. This value must be sent by the toolbar on a new communication event when transferring to the second agent.
Skip customer identification step	SVMCA_BYPASS_IDENTIFY_CONTACT	Skips automatic customer identification upon receiving an inbound interaction. This token can be used with the <code>startCommEvent()</code> call.
Skip customer verification step	SVMCA_BYPASS_CUSTOMER_VERIFICATION	Skips automatic customer verification screen to be made available to the agent. This token can be used with the <code>startCommEvent()</code> call.
Skip Reverse Lookup	SVMCA_BYPASS_REVERSE_LOOKUP	Skip automatic reverse lookup of contact.
Skip the auto screen pop engine	SVMCA_BYPASS_AUTO_SCREEN_POP	Skips the automatic run of screen pop engine upon accepting the interaction by the agent. This token can be used with the <code>startCommEvent()</code> call.
Timeout	SVMCA_WRAPUP_TIMEOUT	Specify the time, in seconds, before the Wrap Up window is automatically closed, by adding the following pair to this parameter: SVMCA_WRAPUP_TIMEOUT, #OFSECONDS. Replace the #OFSECONDS value with the actual number of seconds before the wrap up is automatically closed. This token can be used with the <code>newCommEvent()</code> call.

System Pages

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."

Page Name	Page Code	What does it do
Create Account	Create_Account	Accepts details of an account.

Page Name	Page Code	What does it do
Create Contact	Create_Contact	Accepts details of a contact.
Create Lead	Create_Lead	Accepts details for a lead.
Create Opportunity	Create_Opportunity	Accepts details for an opportunity.
Create Service Request	Create_Service_Request	Accepts details of a service request.
Create Work Order	Create_Work_Order	Accepts details of a work order.
Edit Account	Edit_Account	Displays the details of an account or organization information. The displayed page information can be changed based on the rights associated with agent role.
Edit Asset	Edit_Asset	Displays details of an asset.
Edit Contact	Edit_Contact	Displays details of a contact or customer information. The displayed information can be changed based on the rights associated with agent role.
Edit Contact - Leads	Edit_Contact_Leads_Tab	Displays details of a lead for a given contact or customer.
Edit Contact - Opportunities	Edit_Contact_Opportunities	Displays details of an opportunity for a given contact or customer.
Edit Contact - Service Requests	Edit_Contact_Service_Requests	Displays details of a service request for a given contact or customer.
Edit Lead	Edit_Lead	Displays details of a lead.
Edit Loyalty Member	Edit_Loyalty_Member	Displays details of a Loyalty member.
Edit Opportunity	Edit_Opportunity	Displays details of an opportunity.
Edit Service Request	Edit_Service_Request	Shows details of a service request. Page information can be edited according with the rights associated with agent role.
Edit Service Request - Work Order	Edit_Work_Order_From_SR	Shows details of a work order for a given service request.
Edit Work Order	Edit_Work_Order	Shows details of a work order.

Page Name	Page Code	What does it do
Opportunity List	Opportunity_List	Shows details of an opportunity.
Service Request List	Service_Request_List	Shows details of a service request.
Work Order List	Work_Order_List	Shows details of a work order.

Related Topics

- [Overview of Interaction APIs](#)

System Business Objects

The following table lists the business objects.

What would you like to do?	How to do it
Describe account or organization information.	Select Account
Describe activity information	Select Activity
Describe asset information.	Select Asset
Describe the contact or customer information.	Select Contact
Describe lead information.	Select Lead
Describe loyalty member information.	Select Member
Describe opportunity information.	Select Opportunity
Describe person information.	Select Person Picker
Describe resource information.	Select Resource
Describe service request information.	Select ServiceRequest
Describe work order information.	Select WorkOrder

What would you like to do?	How to do it
Describe wrap up information.	Select Wrap Up

Channels Code

Channels are configured using the Functional Setup Manager within Oracle Service Center. The following table lists the channel codes:

Channel Code	Description
PHONE	Phone channel.
CHAT	Chat channel.
EMAIL	Email channel.

Application Classification Code

The following table lists the ready-to-use application classification codes that the application recognizes. The list of application classifications can be modified using Functional Setup Manager.

Application Classification Code	Description
ORA_HRHD	Default classification for Human Resources Help Desk related setup for Lookup rules and Screen Pop rules.
ORA_SALES	Default classification for sales-related setup for Lookup rules and Screen Pop rules.
ORA_SERVICE	Default classification for service-related setup for Lookup rules and Screen Pop rules.

Channel Type Data

The following table lists the channel types, which are predefined lookups identified by the ORA_SVC_CHANNEL_TYPE_CD lookup type.

Channel Code	What is it used for
ORA_SVC_CHAT	Chat channel
ORA_SVC_EMAIL	E-mail channel
ORA_SVC_PHONE	Phone channel
ORA_SVC_SOCIAL	Social channel
ORA_SVC_WEB	Web channel
ORA_SVC_NONE	None

Call Flow Parameter Settings

Call Flow Parameter Settings

The call flow parameters determine the default and administrator-defined actions to be performed for a customer interaction. Customer interactions are calls with the customer in Sales for Redwood and calls or chats in Service Center. Administrator-defined settings override the default settings.

You can specify the settings listed in the following table:

Setting	Options	Description
Contact Required	Yes/No	Specifies whether the contact details of a customer must be displayed. Note: For Sales for Redwood implementations, you must select No . Select No if you're using Sales for Redwood.
Contact Verification	Yes/No	Specifies whether the customer details must be verified at the beginning of the call. Note: For Sales for Redwood implementations, you must select No .
Screen Pop	Yes/No	Specifies whether a screen pop page must be displayed. Select Yes .

Setting	Options	Description
Wrap Up	Yes/No/Server Driven	Specifies whether a customer interaction requires wrap up. Select the Server Driven option to get notified by the toolbar if an interaction must be wrapped up. Note: For Sales for Redwood implementations, this setting must always be enabled.

Configure Call Flow Parameters

Here's how you configure call flow parameters:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Configure Call Flow Parameters
2. Specify default settings for all interaction channels in the **Default Settings** section.
3. In the **Administrator-Defined Settings** section, specify the settings for the individual interaction channels.
These settings override the default settings. For example, the default setting is to verify the contact.
4. Click **Save and Close**.

Related Topics

- [How do I configure screen pop pages?](#)
- [How do I update existing setup data?](#)

Configure Wrap Up

Populate Wrap Up Parameters Automatically

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

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

To configure a wrap up 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.

Create a Wrap Up Token

To create a wrap up 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's used to represent a token
- **Description:** String to provide extra information about the token
- **Object Name:** Business object to which the token is associated. Select Wrap Up from the drop-down list.
- **Object Attribute:** Attribute of the wrap up business object that's associated with the token. Object Attribute is an optional value.

4. Specify the values for the token and click **Save**.

Related Topics

- [Overview of Sandboxes](#)
- [Create and Activate Sandboxes](#)
- [Publish Sandboxes](#)

Implement timed wrap up

To set up timed wrap up for chat and calls you just need to set the `ORA_SVC_MCA_WRAP_UP_TIMEOUT` profile option.

Here's a few scenarios where timed wrap up is used:

Chat Transfers

Transferring to a queue:

- If timed wrap up is disabled, the Wrap Up is displayed when a chat is transferred to the queue, and the chat capacity is released immediately when the chat ends.
- If timed wrap up is enabled when a chat is transferred to a queue, the Wrap Up screen appears along with a countdown timer. The service representative's chat capacity is automatically released once the timer expires.

Transferring to service representative:

- If timed wrap up is disabled when a chat is transferred to another service representative, the Wrap Up screen appears for the initial service representative. The initial service representative's chat capacity is released immediately once the new representative accepts the chat.
- If timed wrap up is enabled when a chat is transferred to another service representative, the initial service representative's Wrap Up screen and countdown timer appear. The initial service representative's chat capacity is automatically released when the timer expires.

Chat Conferences

Lead service representative ends the chat:

- If timed wrap up is disabled, the Wrap Up is displayed for all service representative chat participants when a chat ends. The chat capacity is released immediately when the chat ends.
- If timed wrap up is enabled when a conference chat ends, all participating service representatives see the Wrap Up screen with the countdown timer. The service representative's chat capacity is automatically released when the timer expires.

Conferee leaves the chat:

- If timed wrap up is disabled, the Wrap Up screen appears when a service representative leaves a chat. Chat capacity is released immediately when the service representative leaves the chat.
- If timed wrap up is enabled, the Wrap Up screen with a countdown timer appear when a service representative leaves a chat conference. The service representative's chat capacity is automatically released when the timer expires.

Here's how you enable the profile option:

1. In Setup and Maintenance, click  and search and select Manage MCA Profile Options.
2. Select ORA_SVC_MCA_WRAP_UP_TIMEOUT, and update the Site profile value to a number greater than zero (0).

The number you choose specifies the number of seconds before the Wrap Up timer expires and chat capacity is released.

Sync Wrap Up outcome and call disposition notes between the media toolbar and Fusion Service

You can optimize the call disposition process by synchronizing the Wrap Up outcome and call disposition notes between the Media toolbar and Service Center having updates reflected in the Outcome and Notes from the Media toolbar to the Fusion Service call panel or the other way around.

Capturing notes as part of the call disposition is a standardized way of documenting the key details of a customer interaction. Improving the call disposition process helps ensure agents see the latest information regardless which UI element they're viewing.

You can extend your application using Oracle Visual Builder Studio to do the following:

- Detect the change during the Call Wrap Up from an external media toolbar, by using the *Field Value Change Event*.
- Set a value for a particular field during the Call Wrap Up from an external media toolbar, by using the *Set Field Value Operation*.
- Get a value for a particular field during the Call Wrap Up from an external media toolbar, by using the *Get Field Value Operation*.
- Save the updates to the Call Wrap Up from an external media toolbar, by using the *Save Record Operation*.

Here's a JavaScript example showing how to get WrapUpRecordContext from CloseComm and perform the Set Field Value operation in Wrap Up:

```
/// <reference path="uiEventsFramework.d.ts"/>
const uiEventsFrameworkInstance = await CX_SVC_UI_EVENTS_FRAMEWORK.uiEventsFramework.initialize('appname',
'v1');
const multiChannelAdaptorContext = await uiEventsFrameworkInstance.getMultiChannelAdaptorContext();
const phoneContext = await multiChannelAdaptorContext.getCommunicationChannelContext('PHONE');
const request = uiEventsFrameworkInstance.requestHelper.createPublishRequest('closeCommEvent');
request.setAppClassification('appClassification');
request.setInputData(_inboundData);
request.setReason('WRAPUP');
const inData = request.getInData();
const response = await phoneContext.publish(request);
const engagementContext = response.getResponseData().getEngagementContext();
const wrapUpRecordContext = await engagementContext.getWrapupContext();
const requestObject =
uiEventsFrameworkInstance.requestHelper.createPublishRequest('cxEventBusSetFieldValueOperation');
requestObject.field().setValue('WrapUp.ResolutionCd', 'ORA_SVC_DS_POSITIVE_OUTCOME');
requestObject.field().setValue('WrapUp.CallNotes', 'New Description');
wrapUpRecordContext.publish(requestObject).then((message) => {
const response = message;
// custom code
}).catch((error) => {
// custom code
});
```

Overview of Extending Wrap Up

Here's a short overview on what you can do to extend the Wrap up UI.

Overview of Extending the Wrap Up UI

1. Open the Oracle Cloud Application page containing the component you want to extend.
2. Click the **Settings and Actions** menu and select **Edit Page in Visual Builder** to open the page in the **VB Studio Page Designer**.

3. Create a project or Open project with base application as: **Oracle CX Service UI Extension APP**.

New Project

1 Details 2 Template 3 Properties Finish

Project Properties Application Extension

Extension Name: wrapupextens.Extension
Extension Id: site_wrapupextens.Extension
Workspace Name: Workspace 1
Oracle Cloud Applications Development Instance: fa-smc17
Git Repository Name: wrapupextens.git
branch!

Base Application
Oracle CX Service UI Extension (from Oracle CX Service UI Extension App)
Oracle CX Service UI factory extension application

4. Click the **Layouts** subtab, then from the Dependencies list click **WrapUps**.
5. In the Dynamic Form list, select the **Wrapup dynamic layout**.

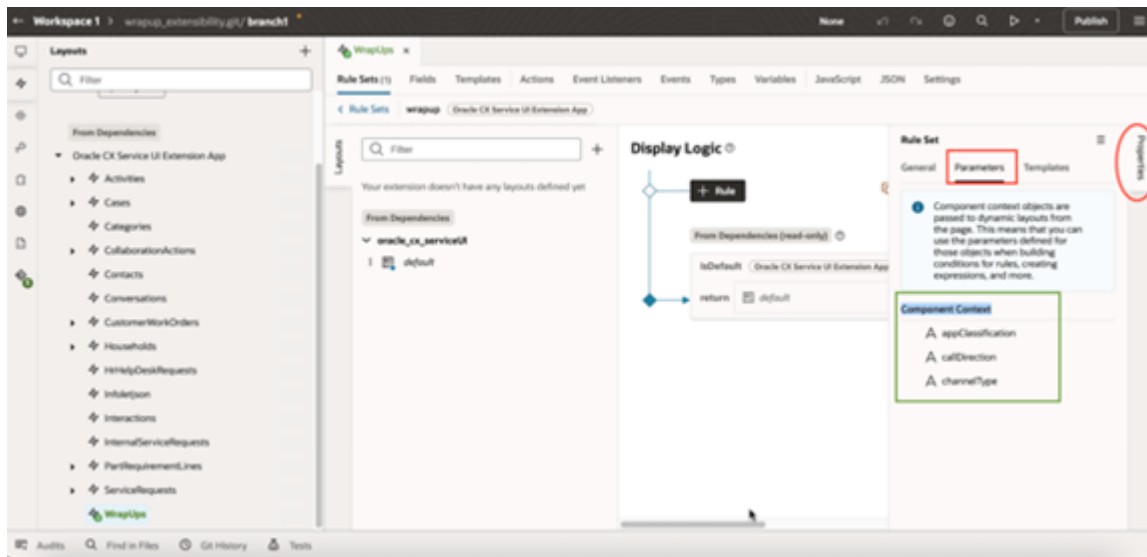
Workspace 1 wrapup_extensibility.git/branch!

Layouts Rule Sets Fields Templates Actions Event Listeners Events Types Variables JavaScript JSON Settings

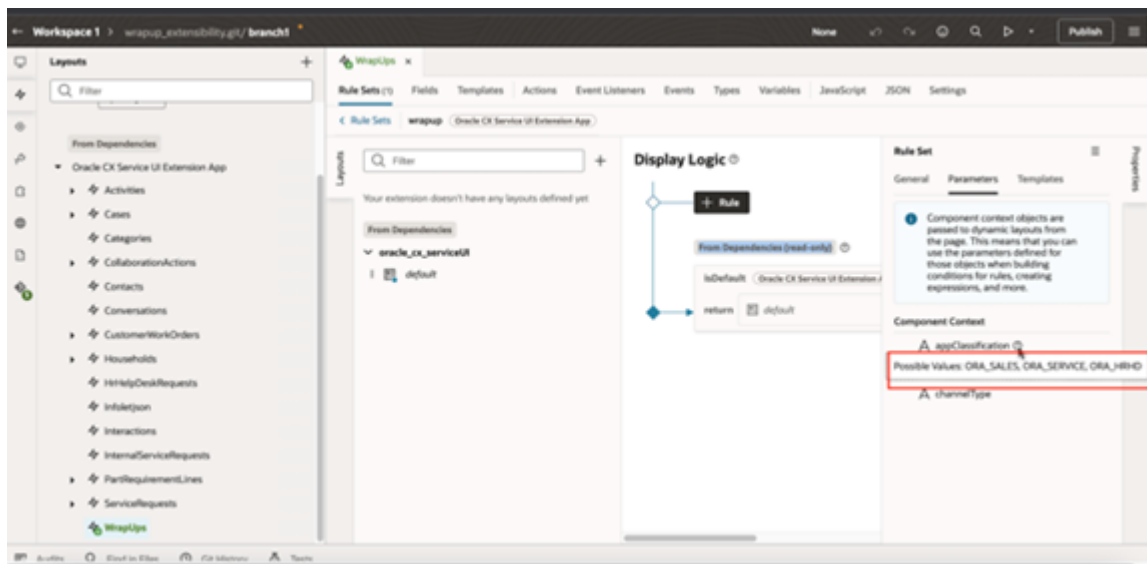
Dynamic Form

wrapup - Oracle CX Service ...
wrapup dynamic layout

6. In the **Rule Set** field, click the **Parameters** tab.

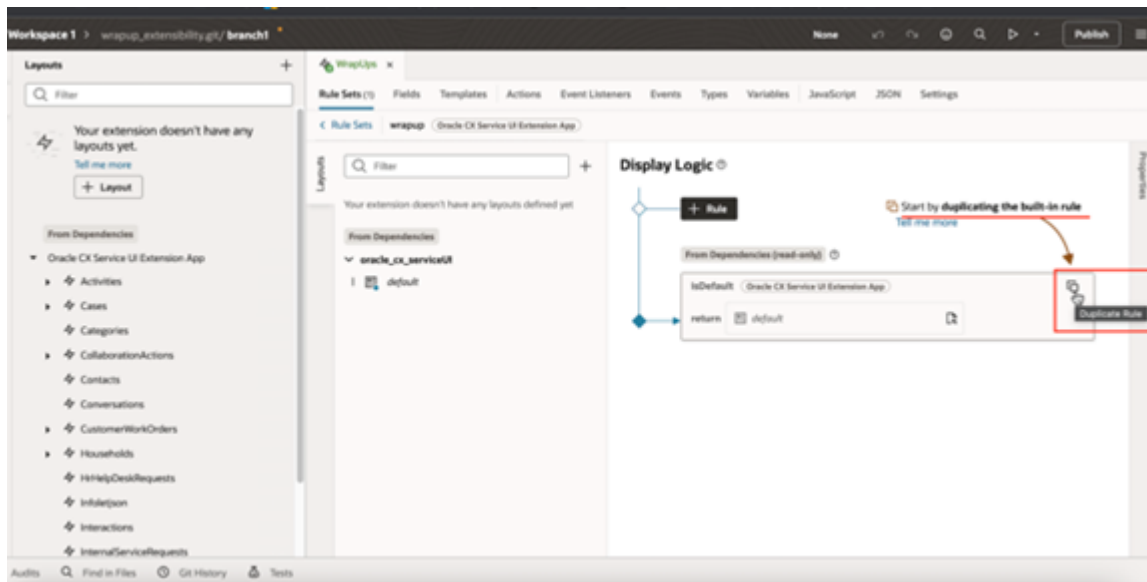


7. The Wrapup component has the following three contexts:
- appClassification
 - callDirection
 - channelType
8. Hover each context to highlight it, then hover of the ? symbol to view possible values for each component context.



9. Possible values are:
- appClassification: Possible Values: ORA_SALES, ORA_SERVICE, ORA_HRHD
 - callDirection: Possible Values: ORA_SVC_INBOUND, ORA_SVC_OUTBOUND
 - channelType: Possible Values: ORA_SVC_PHONE, ORA_SVC_CHAT

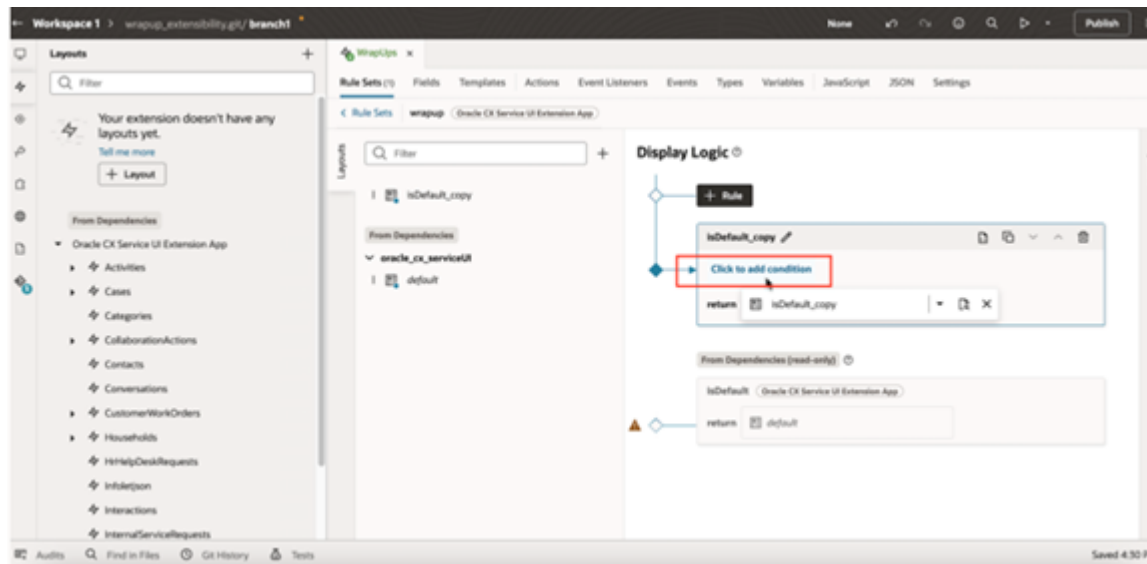
10. In the **Display Logic** area, click the **Duplicate Rule** icon to clone the default rule.



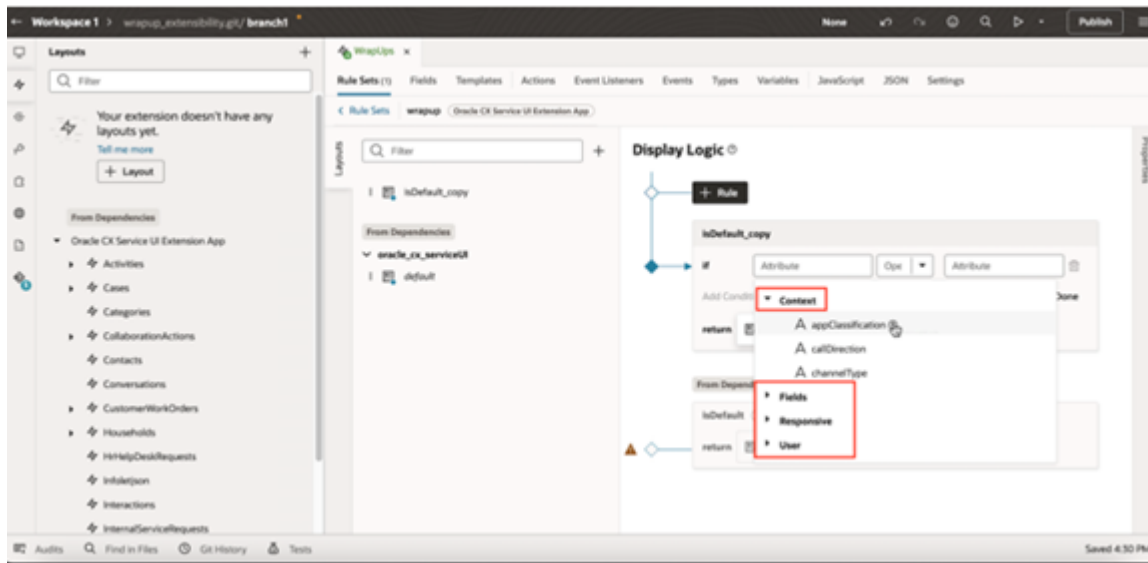
11. Enter a name for the duplicate rule then click the **Duplicate** button to create a copy of rule and layout.

Note: Default layouts from dependencies are read-only, you can't edit them.

12. In the the newly copied layout, select **Click to add condition** to add conditions for when to use this rule and layout.



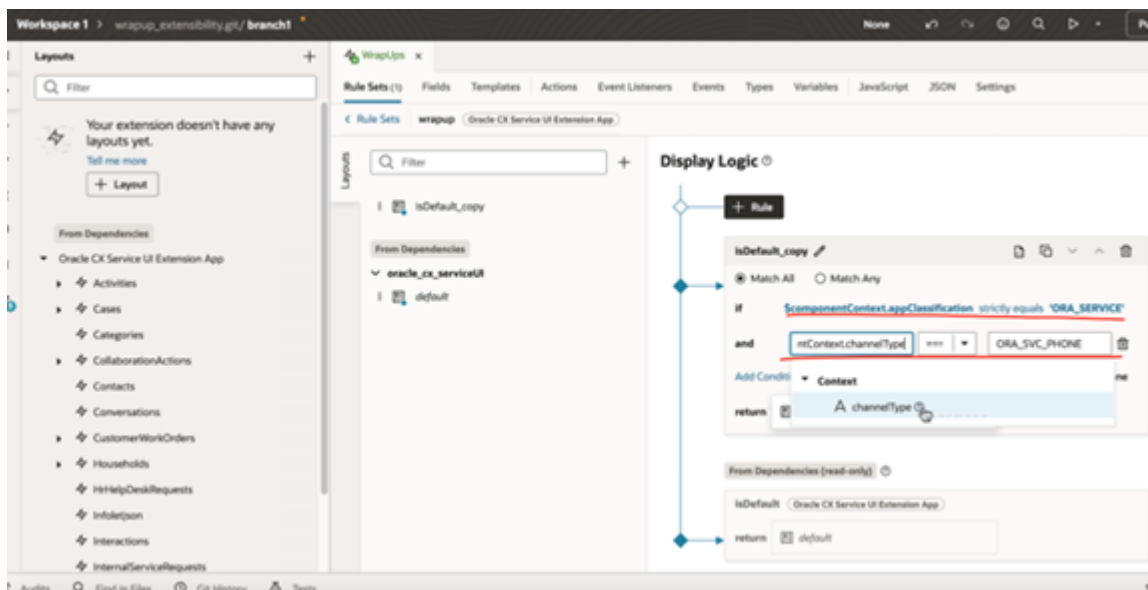
13. You can add conditions as shown here:



14. And you can add them based on the following:

- Context
- Fields
- Responsive
- User

15. You can give multiple conditions, or you can add multiple rules based on your requirement.

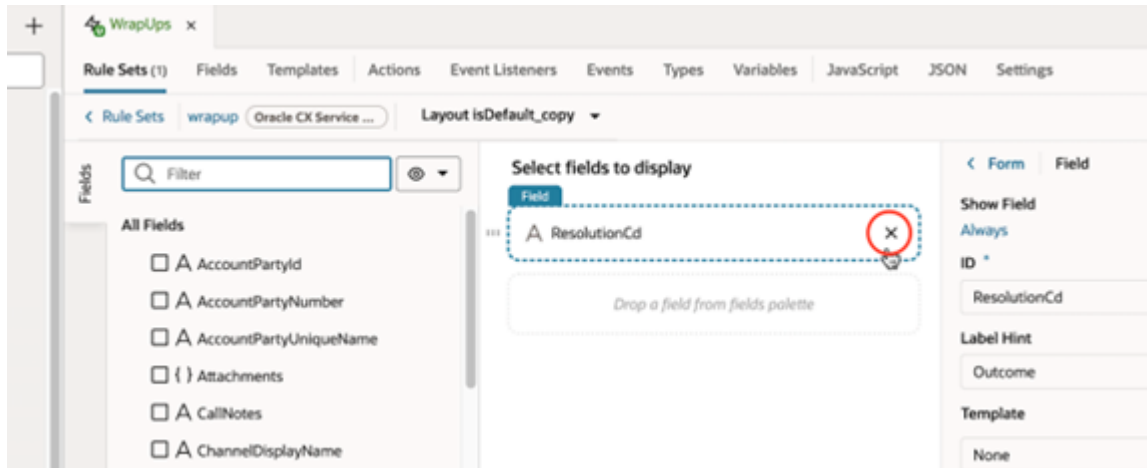


16. Now we'll create rules, so go to the Layouts area, click the **+ Rule** button. For more information on create a rule, refer to [Create a Rule in a Rule Set](#).

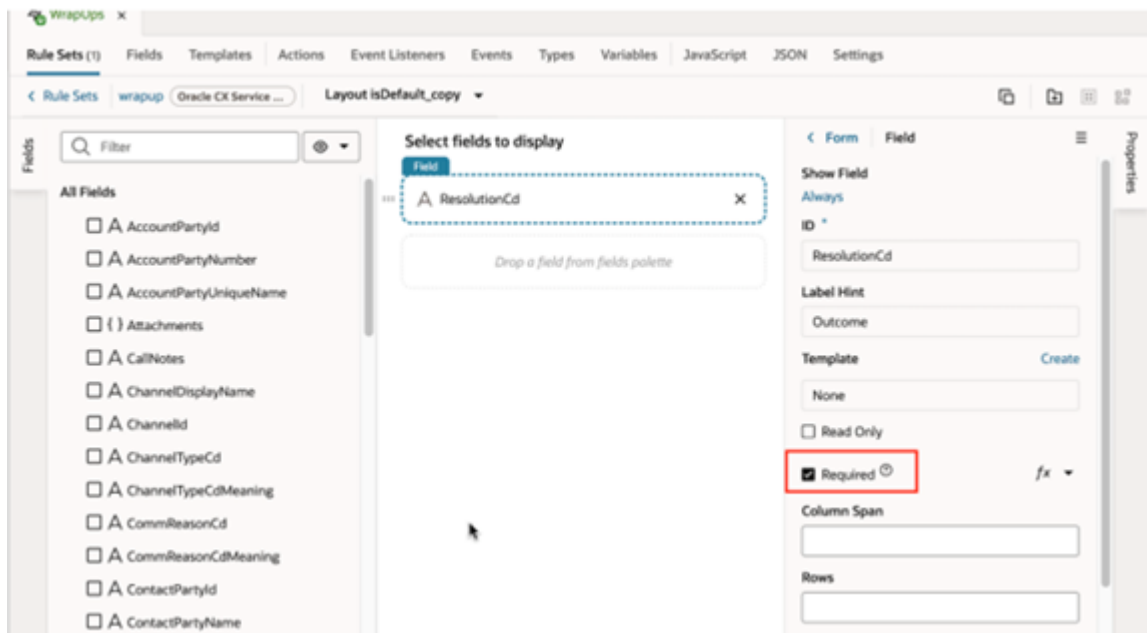
Further Extending You Can Do

For further information on extending the application, refer to these topics:

- To add additional fields to the layout, see [How do I add, hide, or rearrange fields?](#).
- To delete a field, hover over the field, and click the **Delete** button.



- To mark a field as mandatory, click the the **Required** checkbox in the properties for the given field.



- To make fields conditionally required or read only, see *How do I make a field read only or required?*. Use the example with the following inputs:
 - a. First, make a field required conditional by using Function **fx** option of the Required flag.
 - b. Then enter the required key and value. Here are some examples:
 - `[[ $componentContext.callDirection == 'ORA_SVC_OUTBOUND' ]]`
 - `[[ $user.userName == 'ctiautouisvr3' ]]`
 - `[[ $fields.ResolutionCd.value() === 'ORA_SVC_DS_PROVIDED_INFO' ]]`
 - c. Now, to display the field conditionally, select the **Show field** option in the properties section of a field.
 - d. Click **Always** to enter a condition.
 - e. Save to view the updated Show field condition.

14 Set Up Digital Assistant as a Chat Bot

How do I create Oracle Digital Assistant as an agent?

Oracle Digital Assistant (ODA) can be configured to work with your live agents to resolve your end users' issues. You setup your ODA as an agent similar to your live agent setup in Fusion, then some extra setup is necessary in IDCS and the Oracle Digital Assistant applications.

Create your ODA user

1. Navigate to My Team > Users and Roles
2. Click the **+** button to add a new user
3. Enter your ODA agent information:
 - a. Enter last name and first name.
 - b. Enter the email address. This doesn't need to be a valid email address)
 - c. Enter the user name. This will be the ODA sign in name you'll use.
 - d. Deselect **Send user name and password**.
 - e. Select **Employee** for Person Type.
 - f. Select your **Legal Employer**.
 - g. Select your **Business Unit**.
 - h. Select your **Resource Role**, such as Customer Service Representative.
 - i. Click the **Auto Provision Roles** button.
 - j. Click **Save and Close**.

Add the Chat Agent Role

1. Navigate to **Tools > Security Console**.
2. Click the **User** tab.
3. Enter the ODA user name.
4. Select your user.
5. Click **Edit**.
6. Click **Add Role**.
7. Enter **Chat Agent** in the search field.
8. Select the Chat Agent role (ORA_SVC_CHAT_AGENT_ABSTRACT).
9. Click **Save and Close**.
10. Click **Reset Password**.
11. Select **Manually change the password**.
12. Enter a new password, using at least 8 characters and 1 number.
13. Click **Reset Password**.

Note: You must wait a full 24 hours for Oracle Identity Cloud Service to sync your Fusion users and roles. If you need to sync manually, see the following steps.

Create a User Agent for Digital Assistant

Before you get started, make sure you have permissions which will enable you to create resources and queues, such as the Service Request Power User Duty Role.

Then you'll create a chat agent, and validate and add additional roles to that user.

First, you create the new chat user.

1. From the Navigator menu, expand the My Team list, then select the **Users and Roles**.
2. Click the Create (+) icon.
3. Enter the required information from the Personal Details area that match the bot persona.
Note: The Last Name field appears first in the form.
4. From the **Legal Employer** drop-down list, choose your company name, and then from the **Business Unit** drop-down list, select the correct business unit.
5. In the Employment Information, chose **Employee** from the drop-down list.
6. In the Resource Information area, choose **Service Representative** from the drop-down list.
7. Click the **Autoprovision Roles** button.
8. Click **Save and Close**.

Now you validate the new user and add additional roles.

1. Sign in as an Application Administrator.
2. Click Navigator, then from the Tools menu, select **Security Console**.
3. Click the **Users** tab and search for the new user.
4. Select the new user by clicking the user link.
5. Click **Reset Password**, then enter a new password.
6. Then, from the User Account Details page, click **Edit**.
7. In the Roles area, click the **Add Role** button.
8. In the Add Role Membership from Role dialog box, search for the **Chat Agent** abstract role
Note: The role code is ORA_SVC_CHAT_AGENT_ABSTRACT.
9. Click **Add Role Membership**, then click **Done**.
10. Click **Save and Close**.

Configure Chat Options and Enable Notifications

You can skip this step if you've already set up service request routing.

If you haven't configured your chat profile options, refer to [How do I enable features and profile options for chat?](#).

For the ODA integration to work, you must enable the SVC_CHAT_INLAYS_ACCESS_ENABLED profile option.

Set Up Queues and Agent Capacity for the Bot User

We recommend that you start with a capacity of 100 and increase the value based on your peak concurrent chat traffic volume trends.

You can override the global chat capacity for the bot to 100 vs. modifying the global capacity. However, this number shouldn't exceed your max allowed concurrent ODA sessions. You can always augment this live agent assistance using overflow queues should you need additional capacity than your max allowed concurrent ODA sessions.

Note: As a prerequisite to this task, you must enable the Edit Capacities option.

Enable the Edit Capacities Option

1. Click Setup and Maintenance.
2. From the Setup page, choose **Service**, and then from the Functional Areas list, select **Communication Channels**.
3. Click the **Manage Capacities** task.
4. Make sure the **Enable Channel Capacities to Be Overridden for Individuals** checkbox is selected.
5. Click **Save and Close**.

Set Up the Chat Bot Queue

1. From the Service homepage, click the **Queues** tile to view the Queues dashboard.
2. Click **Create Queue**.
3. Set up your bot queue by doing the following:
 - a. Enter a name for the queue.
 - b. For the Distribution field, click **Automatic**.
 - c. Click **Save and Continue**.
 - d. Click the **Resources** icon in the explorer.
 - e. Click the **Add Resources** button, then search for the bot you created.
 - f. Select the bot and click **Apply**, and then click **OK**.
 - g. Click **Save and Close**.
4. From the queues dashboard click the **Capacities...** button.
5. On the **Edit Capacities** workspace, click the **Agents** drop-down list, and choose **Select Individuals**.
6. On the **Add Agents** workspace, enter your chat bot name, then click **Search**.
7. Select the chat bot you created from the search results, and click **OK**.
8. On the **Edit Capacities** workspace, set the capacity of the Chat field to a high number, and then click **Save and Close**.

Related Topics

- [Add Resources to a Queue](#)

Extend the Universal Work Object

You'll extend the universal work object by adding a custom field. This task is needed so you can differentiate ODA routing from the live agent routing.

1. Sign in as an administrator.
2. Click **Navigator > Configuration > Sandboxes**. You can either create a new sandbox, or use an existing one and set it to active.

For information about sandboxes, refer to [Overview of Sandboxes](#).

3. Under Configuration, choose **Application Composer**.
4. In the explorer, expand Standard Objects, then expand the **Universal Work Object**.
5. Click Fields, then click the **Create a custom field** icon.

For this example, we're going to use Escalation level.

6. Select **Number** for Field Type, and click **OK**.
7. On the Create Number Field work area, enter **Escalation Level** in the **Display Label** field, and then click **Save and Close**.
8. Click the drop-down list for your sandbox name, and choose **Publish**.

After you've created the new customer attribute on the Universal Work Object, and published the sandbox, you then expose the new object using the the Manage Service Assignment Objects task.

9. Select **Setup and Maintenance** from the Settings and Actions menu, and select the following:
 - Offering: Service
 - Functional Area: Communication Channels
 - Task: (select All Tasks from the Show menu) Manage Service Assignments Objects
10. Select the **Any** object to extend.

This is the Universal Work Object.

11. In the Details area, click the **Attributes** tab.
12. Click the Add Row (+) icon.
13. Click the **View Object Attributes** drop-down list and select the custom field you created in the previous steps (Escalation Level (EscalationLevel_c)).
14. Click **Save and Publish**.

How do I create routing rules?

Routing rules are used to identify queue assignments which could be to a queue with live agents or the ODA queue. They're also used to escalate a chat from ODA to a live agent queue.

Note: You can use routing rules to route chats to either ODA as an agent or to a live agent. Bot queue assignment rules must be created in the Generic Queuing Rules Category. Here's how you create the rule set:

1. From the Settings and Actions menu, select **Setup and Maintenance**, and select the following:
 - o Offering: **Service**
 - o Functional Area: **Communication Channels**
 - o Task: (select All Tasks from the Show menu) **Manage Service Assignment Rules**
2. From the **Category** drop-down list, select **Generic Queuing Rules**.

Note: Depending on what application you're using, Service Center, Help Desk, and so on, you select the corresponding generic category.

3. In the **Rule Sets** area, click the Add Row (+) icon.
4. Enter a name for the rule set.
5. In the **Rules** area, click the Create (+) icon, then do the following:
 - a. Enter a name and description in the respective fields.
 - b. (Optional) Set an effective start and end date.
 - c. In the Conditions area, click the Add Row (+) icon.
 - d. From the **Object** drop-down list, select **Any**.
 - e. From the **Attributed** drop-down list, select **Escalation Level**.
 - f. Click the **Operator** drop-down list, select **Equals**, and then in the Value field, enter 1.

Note: If you're setting up your chat requests to route to Oracle Digital Assistant, you'd set the Escalation Level to equal "0" (zero), then you'd create a second rule within the same rule set to route to a live agent queue. The second rule would have the Escalation Level be greater than 0 (zero).

You can add other criteria to use for routing as needed.

Note: Rules must be unique. The new attribute you created in the Universal Work Order object and exposed to assignment manager (Name: EscalationLevel_c, Type: Number, Default Value: 0) is intended to break a tie or ensure that rules are unique.

6. When you're finished, click **Save and Publish**.
7. To confirm that the rule has been published, click the **Refresh** button.

A few things to note with routing rules

- **Custom Attributes:** If you've added a custom attribute and you want to use it in routing rules, then you must expose the custom field or object using the Communication Channels > Manage Service Assignment Objects page task on Functional Setup Manager.
- **Scoring routing rules:** 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 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 didn't 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.

How do I synchronize users and roles to Oracle Identity Cloud Service?

All Fusion users and roles are synchronized with Oracle Identity Cloud Service. That is, they exist in both Oracle Identity Cloud Service and Fusion Service applications. A scheduled process job synchronizes Fusion users with IDCS. However, if you don't want to wait for the job to run you can follow the steps outlined in this task.

There are two parts to performing a sync between your Fusion Service application and Oracle Identity Cloud Service. The first part must be done in the Fusion Service application. Make sure that your Channel Service (Chat) application is associated with your Oracle Identity Cloud Service.

Note: You must do the following after changing any users or roles.

- Import User and Role Application Security Data
- Run the following two synchronization jobs. These jobs sync both users and roles.
 - *Retrieve Latest LDAP Changes* from the Securing Applications guide.
 - Send Pending LDAP Requests. See *Why You Run the Send Pending LDAP Requests Process* topic in Securing Sales and Fusion Service.

To run the jobs, follow these steps:

1. Navigate to **Tools > Scheduled Processes**.
2. Click **Schedule New Process**.
3. In the Schedule New Process window, click the **Name** drop-down list, and select **Search...**
4. Copy and paste **Import User and Role Application Security Data** in the Name field and click **Search**.
5. Select the returned row and click **OK**, and then click **OK** again.
6. In Process Details window, click **Submit**.
The Import User and Role Application Security Data job will be run.
7. Wait until the job succeeds before proceeding.
8. Repeat steps 2-7 to run the job **Retrieve Latest LDAP Changes**.

9. Repeat steps 2-7 to run the job **Send Pending LDAP Requests**.
10. Click **Save and Close**.

Link with Oracle Identity Cloud Service

To perform setup in Identity Cloud Service, you must login using a user that has the Identity Domain Administrator role

Using a new browser or new incognito window, log in to Oracle Identity Cloud Service (IDCS) as the administrator user. You may leave your service application screen where you are currently and come back to verify connectivity once this step is completed.

Now you complete the tasks which follow:

Create a Resource Application

Configure a Client Application

Create a Resource Application

You now have to create an Oracle Identity Cloud Service (IDCS) Resource Application for the chat service. ODA will use this service to create an IDCS Client Application for requesting JWT tokens.

1. Log in to Identity Cloud Service as a user with the Identity Domain Administrator role.
2. Click the Navigation Drawer icon to show the side menu.
3. Select **Applications**.
4. From the Applications screen, click **+Add** and select **Confidential Application**.
5. Enter a name for the resource, such as Channel Service Resource, and then click **Next**.
6. On the Client screen, select **Configure this application as a client now** and select the following two checkboxes in the **Allowed Grant Type** area:
 - Resource Owner
 - Client Credentials.
7. Click **Next**.
8. On the Resources screen, select **Configure this application as a resource server now**.
9. In the Primary Audience field, enter the URL for the Chat VIP where the chat service is deployed.

Note: At the end of the URL, append the following: /engagement/api. For example: https://<FQDN of Chat Service> /engagement/api. The configuration API will contain the actual information required to make the connection.

10. In the **Configure application APIs that need to be OAuth protected** workspace, click the Scope **Add** button.
11. In the **Add Scope** pop up window, enter **/agent** in the Scope field, then click the **Add** button.
12. Click **Next**, and then click **Next** again.
13. Click **Finish**.
14. On the pop up window, copy the **Client ID** value and the **Client Secret** value for later use.
15. Close the pop up window, and click **Save**.
16. Click the **Activate** button.

Note: You must click the **Activate** button before closing IDCS.

Configure a Client Application

Once you've configured a Resource application, you must create a Client application that uses the Resource. The Client application is used by ODA to request a JWT to send to the Chat Service for authentication. Creating a Client application is similar to creating a Resource application.

1. Log in to Identity Cloud Service as a user with the Identity Domain Administrator role.
2. Click the Navigation Drawer icon to show the side menu.
3. Select **Applications**.
4. From the Applications screen, click **+Add** and select **Confidential Application**.
5. Enter a name for the Client application (such as Channel Service Application) and click **Next**.
6. On the Client screen, select **Configure this application as a client now** and select the following two checkboxes in the **Allowed Grant Type** area:
 - o Resource Owner
 - o Client Credentials.
7. Click **Next** to advance to the **Resources** tab.
8. In the **Token Issuance Policy** section, click the **+Add Scope** button.
9. In the Select Scope pop up window, locate the resource you previously created and click the **>** arrow beside it. The Select Scope page shows a list of the defined scopes for the resource.
10. Click the checkbox, and then click **Add**.
11. Click **Next** until you reach the Authorization workspace, and then click **Finish**.
12. On the pop up window, copy the **Client ID** value and the **Client Secret** value for later use.
13. Close the pop up window, and click **Save**.
14. Click the **Activate** button.

Note: You must click the **Activate** button before closing IDCS.

Provision Chat Service

Once you've completed the preceding steps, contact Oracle Support for assistance in completing the Oracle Digital Assistant provisioning process.

Configure Oracle Digital Assistant

In Oracle Digital Assistant, you create a DA as Agent channel that associates your digital assistant with the digital-assistant agent. When you enable the channel, the digital assistant is signed in as the digital-assistant agent and is ready to handle chat requests. When you complete this task initiated requests will route to ODA.

When configuring ODA as a digital assistant, follow the steps in *DA as an Agent*.

You'll perform these steps to complete the setup:

- Create or update a Digital Assistant
- Create or extend a Skill.
- Create an Authentication Service for the Channel Service in IDCS
- Create a Digital Assistant as Agent channel
- Add the Digital Assistant to the channel

The ODA Channel that you set up will use this user to authenticate with IDCS and get the token to connect to chat service. To do this the currently synced user from Fusion Service should be made Federated = False and a new password should be assigned.

Here's how you make a user non-federated:

1. From the IDCS homepage Navigate to **Users**.
2. Locate the new user after the Fusion batch job has synced new users to IDCS.
3. Select Federated = **No**.
4. Make sure the user's email is a valid one that you can access.
5. Click the **Reset Password**.
A password reset link is emailed to the email address of the user.
6. Enter a new password complying with your policies

Related Topics

- [Configure the DA as an Agent](#)
- [Cheat Sheet for Building Great Digital Assistants](#)

Create and Extend an ODA Skills

Within the Flow Editor you enter skills which will enable routing of a chat to a human agent, or the bot which can display articles to the user.

For the cases where your customer needs to chat with a human, you use the `system.AgentTransfer` component to transfer the conversation to an agent.

If you're using Oracle Fusion Service Knowledge Management, then you can use the `system.KnowledgeSearch` built-in component to search for and display articles from that service.

You can design these components as separate skills and bundle them together under a Service Digital Assistant or they can be part of the same skill.

When building the Skill modify the `system.AgentTransfer` to include the `customProperties` setting as shown in the following example:

```
doTransfer:
  component: "System.AgentTransfer"
  properties:
    maxWaitSeconds: "300"
    allowTransferIf: "agentSessionsAreAvailable"
  # Example of passing a custom property to Oracle FusionService
```

```
customProperties:
# This is a checkbox custom field in the Universal Work Object.
# Checkboxes take the value of Y (selected) or N (unselected).
- name: "EscalationLevel_c"
value: "1"
acceptedMessage: "The conversation has been transferred to a live agent."
waitingMessage: "I'm transferring you to a human. Hold tight"
rejectedMessage: "Looks like no one is available. Please try later"
errorMessage: "We're unable to transfer you due to a system error."
transitions:
actions:
accepted: "reset"
rejected: "handleRejected"
error: "offerMoreHelp"
next: "reset"
```

Create a New Authentication Service

Now you create a new authentication service.

Here's how:

1. In Oracle Digital Assistant, open the side menu.
2. Click **Settings > Authentication Services**.
3. Click **+ Service**.
4. Select the Authorization Code grant type, and then enter the following values:

- a. **Identity Provider:** Choose Oracle Identity Cloud Service
- b. **Name:** A name to identify the service, such as IDCSOAuthForChat.

Note: Make sure you use the same IDCS hostname that you configured when you specified the **Token Endpoint URL** and the **Authorization Endpoint URL**.

- c. **Token Endpoint URL:** Use the value from your previous IDCS setup.
- d. **Authorization Endpoint:** Use the value from your previous IDCS setup.

Note: Use the **Client ID** and **Client Secret** that you created for the Client application, not the Resource application

- e. **Client ID:** Use the value from your previous IDCS setup.
- f. **Secret:** Use the value from your previous IDCS setup.
- g. **Subject Claim:** Sub
- h. **Refresh Token Retention Period:** 7 days

Note: All optional fields can be ignored.

Set Up Digital Assistant as an Agent in Oracle Digital Assistant

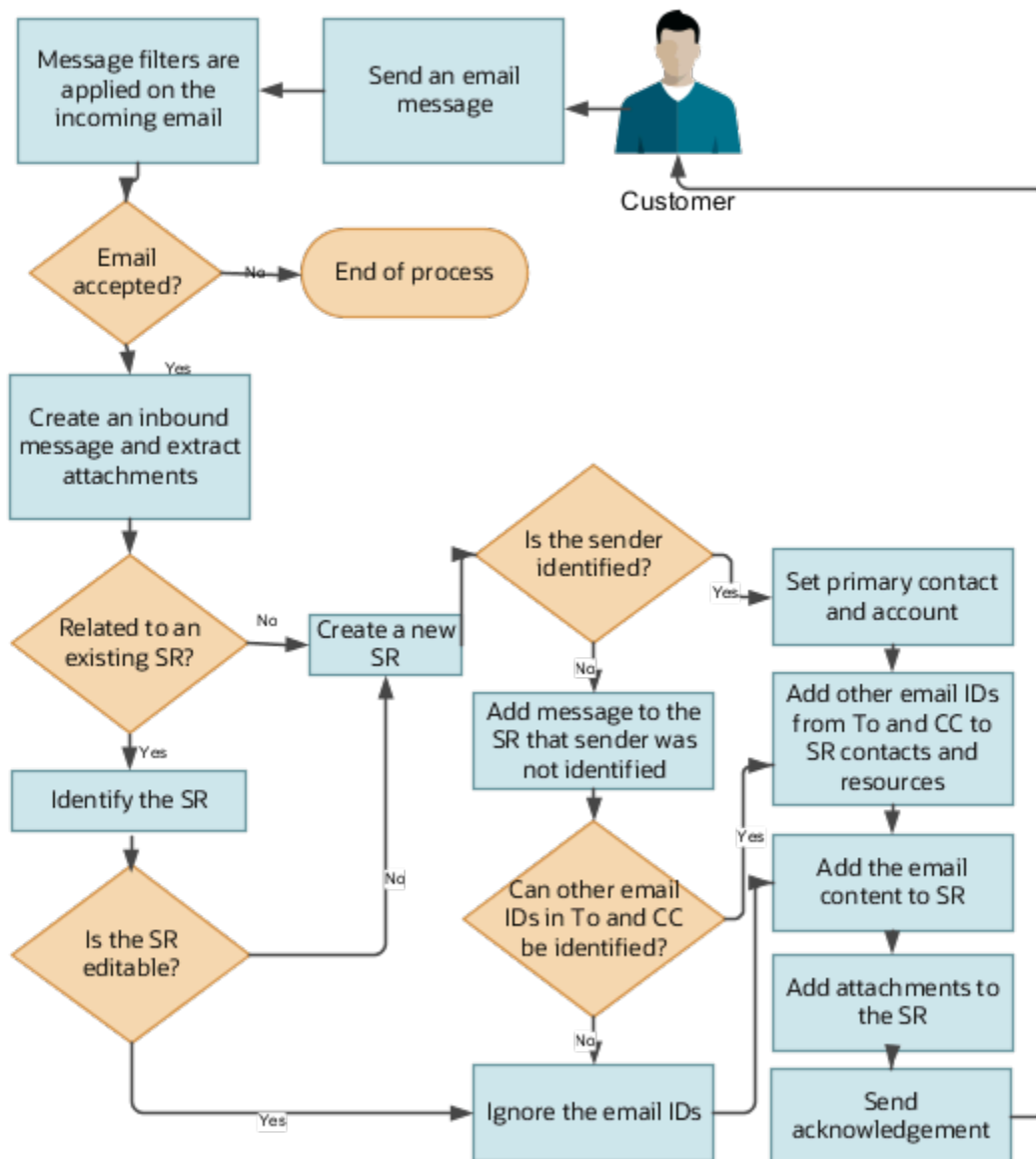
In Oracle Digital Assistant, you create a DA as Agent channel that associates your digital assistant with the digital-assistant agent. When you enable the channel, the digital assistant is signed in as the digital-assistant agent and is ready to handle chat requests.

1. In ODA, click DA as Agent, and then click **+ Add Agent Channel**.
2. Enter a name and an optional description for this channel.
3. Select the Fusion Service
4. Authentication Service: Use the Service created in the previous step
5. Enter the host URL for your Fusion Service instance (fully qualified domain name only).
6. Enter the user name and password (reset via IDCS) for the digital-assistant agent.
7. Click Create
8. In the Route To drop-down field, select the digital assistant to route to.
9. Set the interaction enabled flag to true to log your DA Agent to the chat service

15 Email Channels

How are inbound messages processed?

An inbound message is created when you receive an incoming service email from a customer or a partner . This flowchart describes the process flow for an incoming customer email:



1. A customer or a partner sends an email requesting support.
2. If an inbound message filter exists, then the filter is applied to the incoming message.
3. If the message is accepted, then an inbound message is created, and any associated attachments are extracted.

4. The message is verified to see if it's 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, then the email content and attachments are added to the SR. If the original SR can't be edited, then a new SR is created.
 - If the inbound message isn't related to an existing SR, then 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 the records, then the primary contact and the account are updated on the SR.
 - If the From email ID doesn't exist in the database, then a message is added to the SR that the sender can't be identified.
 - If an SR is created with no contact or account populated.
 - If more than one contact with the same email ID exists in the database, then the following scenarios are possible:

Scenario	Primary Contact in the SR	Acknowledgment Sent
Sender's email address matches the email address of 1 contact, 1 partner, and 1 resource.	The matching contact is set as the primary contact in the SR.	An acknowledgment email is sent to the sender.
Sender's email address matches the email address of 2 contacts.	No primary contact is set in the SR.	An acknowledgment email is sent to the sender. The email states that the sender couldn't be identified because multiple contacts were found with the email address of the sender. More information is requested from the sender.
Sender's email address matches the email address of 2 contacts and 1 partner.	No primary contact is set in the SR.	An acknowledgment email is sent to the sender. The email states that the sender couldn't be identified because multiple contacts were found with the email address of the sender. More information is requested from the sender.
Sender's email address matches the email address of 1 partner and 1 resource.	The matching partner is set as the primary contact in the SR.	An acknowledgment email is sent to the sender.
Sender's email address matches the email address of 2 partners and 1 resource.	No primary contact is set in the SR.	An acknowledgment email is sent to the sender. The email states that the sender couldn't be identified because

Scenario	Primary Contact in the SR	Acknowledgment Sent
		multiple contacts were found with the email address of the sender. More information is requested from the sender.
Sender's email address matches the email address of 1 resource.	The matching resource is set as the primary contact in the SR.	An acknowledgment email is sent to the sender.

6. All other email IDs in the **To** and **CC** fields in the message are checked to see whether they exist as resources or contacts in the application.
 - If they exist as contacts, they're added to the SR contacts.
 - If they exist as resources, they're added to the SR team.

Note: If the other email IDs don't match any resource or contact in the application, then those IDs aren't added to the SR. But they're displayed in the SR message.

7. If the primary contact set in the SR is associated with only one account, then that account is associated with the SR. If the primary contact is associated with more than one account, then the primary account of the primary contact is set as the account on the SR.
8. The incoming message content and the attachments are added to the new SR.

How do I set up inbound and outbound email?

You can configure email to use a single email channel to handle both inbound and outbound email messages. Inbound messages are the messages that you receive from your customers. Outbound messages are the messages that you send to your customers.

Note: When your customers receive an email from your company, they can reply directly to the email. They don't have to change the **To** address in the reply. The To address already displays the account name field of the channel that the agent uses in the outbound message.

Use the following tasks in Setup and Maintenance to set up inbound and outbound email.

No.	Task	Describes how to
1.	Enable Configuration of Email Communications	Enable the tasks required to configure inbound and outbound email communications for service requests.
2.	Configure an Email Channel	Configure an email channel for your organization.

No.	Task	Describes how to
3.	(Optional) Create and Update Inbound Message Filters	Set up 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.
4.	Configure Profile Options for Inbound and Outbound Email	Configure inbound and outbound email profile options.
5.	(Optional) Define Email Templates	Define email templates.
6.	(Optional) Modify Acknowledgment Messages for Inbound Email	Modify the predefined acknowledgment messages provided by the application according to your company's requirement.
7.	Access Point Setup for Inbound Emails: This section consists of multiple topics.	The procedures in this section describe how to verify your email channel configurations and how to register the correct access points with the UMS.
8.	Configure a Job to Process Inbound Emails	Configure a job to process inbound emails to retrieve emails from the customer at regular intervals.

Hide the message identification number in response emails

When Agents are exchanging emails with a customer, a message identification number is automatically stamped on the email to ensure that a reply to that email message is associated with the correct service request.

If this number is distracting to customers or agents, you can hide it in response emails. While still technically on the email message, it isn't visibly rendered. Here's how you enable:

1. In Setup and Maintenance, click the **Tasks** icon, then **Search**.
2. Search for and select **Manage Administrator Profile Values**.
3. Enter ORA_SVC_HIDE_OUTBOUND_MSG_TAG in the Profile Option Code field.
4. In the Profile Values section, specify **Site** level, then select **Yes** from the Profile Value drop-down list.
5. Click **Save and Close**.

Enable Configuration of Email Communications

Before you start your configuration, you must enable the tasks required to configure inbound and outbound email communications for service requests.

1. Sign in as an administrator or a setup user.
2. Navigate to the **Setup and Maintenance** work area and select the **Service** offering.
3. Click the **Change Feature Opt In** link.

The Opt In: page is displayed.

4. In the **Communication Channels** row, click **Edit** in the **Features** column.

The Edit Features: Communication Channels page is displayed.

5. For the **E-mail Communications** feature, select the **Enable** check box.
6. Click **Done**.

How do I configure an email channel?

To send emails to your customers and to receive emails from them, you must first set up an email channel.

You can create separate email channels for different application stripe codes. For example, one for CRM and one for HCM. However, you can use the same email channel for both inbound and outbound emails:

Note: You can add attachments totaling 10MB to an email.

- **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 Oracle's inbound email ID. This is the same email account that Oracle provided at the time of provisioning. For example, all the support emails that are sent to `TechSupport@mycompanydomain.com` are forwarded to `pod_name.fa.extservice.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.

If it's required for your company, you can also create different support email channels for different business units or divisions. For example, `TechSupportDiv1@mycompanydomain.com`, `TechSupportDiv2@mycompanydomain.com`,

and so on. All the support emails sent to these different support email channels are forwarded to `pod_name.fa.extservice.incoming@pod_name-opcwf.mail.dcsn.oraclecloud.com` for processing.

Note: If you're implementing HR Help Desk or Internal Help Desk, then you must define a channel with the appropriate application stripe. The account name of the channel must match the email address of your mailbox from which you're redirecting or forwarding emails to the Oracle mailbox mentioned above. If you don't define the appropriate channel, the requests created for the emails are set with the CRM application stripe. In such cases, you must run an ODI import process to update the application stripes of the SRs.

The SVC_INBOUND_EMAIL_ADDRESSES profile option contains two email IDs:

- For CRM, use the email ID that contains the text 'extservice' as a part of the ID, as the forwarding email ID.

Note: You must avoid updating these inbound email addresses. If you need to update these inbound email addresses for some reason, you must complete the following steps:

- Update the corresponding forwarding rule.
- Ensure that you register the updated inbound email addresses as access points for inbound email. See [How do I register or unregister access points for inbound email?](#)

- 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, you must do the following:

- Set up an SPF policy on your 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.oraclecloud.com ~all.`
- To set the support agent's name as the **From Name** in outbound emails, set the value for the SVC_USE_RESOURCE_NAME_IN_OUTBOUND profile option to **Yes**. For more information, see [How do I set the From Name in outbound emails?](#)

To configure an email channel for the customer:

- In the Setup and Maintenance work area, go to the following:

- Offering: Service
- Functional Area: Communication Channels
- Task: Manage Communication Channels

- In the **Service Channels** page, click **Create Channel**.

- In the **Create Channel** window:

- Select a **Stripe Code**.

Select **CRM** to process emails from and to external customers. Select **HCM** to process emails from and to internal employees, through the HCM Help desk support.

- The **Purpose** field is set to **Support** by default.

- c. Select the **Channel Type** as **Email**.
- d. Specify the support email ID of your company as the **Account Name**. For example:
`support@mycompanydomain.com`

If a forwarding rule is configured, all the mails that are sent to the specified support email ID are forwarded to 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 your behalf.

- e. 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 auto-generated channel code is unique, you can leave it unchanged.
- If the auto-generated channel code isn't unique, add a set of characters to the code to make it unique.

- f. (Optional) 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.
- g. When a new channel is being created, it's active by default. To deactivate it, clear the **Active** option.
- h. (Optional) From the **Business Unit** drop-down list, select a Business Unit (BU).

The BU set in the scope is selected by default, but you can select a different BU. This column appears only if the multiple business units feature is turned on.

- i. Click **Save**.

Inbound Message Filters

How do I use inbound message filters?

Based on the inbound message filter criteria you set, incoming messages from customers or partners can be accepted or rejected.

Message filter patterns are specified using regular expressions. For example, to specify a filter pattern to accept all emails from Oracle employees, the regular expression is given as: `^[A-Za-z0-9._%+~]*@oracle\..com`

You can set more than one filter. Filters are arranged in the descending order of priority on the Manage Email Filters page. The topmost filter has the highest priority. When a message comes in, the filter criteria are checked in the order of priority. If the first filter criterion doesn't apply to a message, the next ones are checked. If any of the filter criteria match the incoming message, the message is accepted or rejected based on the filter specifications.

Note: It's important that you pay special attention while setting up email filters. If the User Messaging Service (UMS) setup rejects any email due to the filter setup, the rejected email can't be processed again by the application and it's lost.

To view the existing inbound message filters or to reorder their priorities:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Email Filters

The Inbound Message Filters page displays a list of the existing inbound message filters.

2. To change the priority of a filter:
 - a. Select the filter whose priority you want to change.
 - b. Click anywhere on the empty spaces in the selected row, and then drag the row to the required position.

How do I create an inbound message filter?

To create an inbound message filter:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Email Filters
2. On the **Inbound Message Filters** page, click **Create**.
3. On 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:
 - o File attachment: A filter is applied to the file type of an attachment.
 - o Header: A filter is applied to a message header.
 - o Mime attachment: A filter is applied to the mime type of an attachment.
 - o Reply to: A filter is applied to the Reply To address.
 - o Sender: A filter is applied to the sender of the message.
 - o Subject: A filter is applied to the subject of the message.

Note: File attachment and Mime attachment filters prevent attachments from getting added to a Service Request (SR) or SR message. But they don't prevent the creation of the SR or SR message.

4. Specify a **Field Name**. The field name is required only for Header filters.
5. Specify an alphanumeric **Filter Pattern**. For examples, see "Examples of Inbound Message Filters".
6. Select an **Action on Pattern Match** to Accept or Reject.
7. Specify an optional **Description**.
8. Click **Create**.

Note: All filters are enabled by default. To disable a filter, update the properties as described in "Update an Inbound Message Filter."

Update an Inbound Message Filter

To update an inbound message filter:

- 1. In the Setup and Maintenance work area, go to the following:
- 2. On the **Inbound Message Filters** page, click the filter type that you want to update.
- 3. On the **Update Message Filter** page, update the filter properties. You can update all 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**.

- 4. After you modify the filter properties, click **Save and Close**.

Related Topics

- [How are inbound messages processed?](#)
- [How do I use inbound message filters?](#)
- [How do I create an inbound message filter?](#)
- [What are some examples of inbound message filters?](#)
- [How do I update existing setup data?](#)

What are some examples of inbound message filters?

Here are some examples of regular expressions for filter patterns of inbound message filters.

Note: Use the "." wildcard characters to match 0 or more occurrences of any character.

Filter Type	Regular Expression Example
File attachment	Attachments that are files with either .pdf, .txt, or .html file extensions: .pdf .txt .html
Header	Emails with headers that contain the string Gentle Reminder: • ^Gentle Reminder[A-Za-z0-9._%+~]* • ^Gentle Reminder.*
Mime attachment	Content Type header with values such as text/plain, text/html, image/jpeg, or application/octet-stream: text/plain text/html image/jpeg application/octet-stream
Reply to	Emails that are sent by the support team:

Filter Type	Regular Expression Example
	- support_[A-Za-z0-9._%+~]*@company\.com - support_.*@company\.com
Sender	Emails that are sent by an Oracle employee: [A-Za-z0-9._%+~]*@oracle\.com .*@oracle\.com
Subject	Email subjects with string AutoReply: ^Auto Reply: [A-Za-z0-9._%+~]* ^Auto Reply.*

How do I configure profile options for inbound and outbound email?

You can set options for incoming and outgoing emails by configuring email profile options. For example: In the SVC_OUTBOUND_EMAIL_FROM profile option, you can set the From email ID that you use to respond to a customer's email.

Configure the inbound email profile options as specified in the following table.

Inbound Profile Options	Description
SVC_ENABLE_INBOUND_EMAIL_DEFAULT_PROCESSING	Indicates whether 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.
SVC_INBOUND_EMAIL_ADDRESSES	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. You must use the address to set a forwarding rule. Try to avoid updating these inbound email addresses. If you update these inbound email addresses for some reason, you must complete the following steps: - Update the corresponding forwarding rule. - Ensure that you register the updated inbound email addresses as access points for inbound email.
SVC_INBOUND_ENABLE_INLINE_ATTACHMENTS	Enables the display of images inline within the SR messages in the UI. The default value is Yes.
SVC_INBOUND_EMAIL_MAX_ATTACH_SIZE	Indicates the maximum size in MB of attachments that are allowed in an inbound email. The default value is 10 MB.

Inbound Profile Options	Description
	Here are some key points to note: - The recommended maximum limit for the email size is 15 MB. Email size includes email headers, attachments, and the email content. Note: You can increase the size of the attachments past 15 MB, but this isn't recommended as it might cause server issues - If an email contains non text content or attachments, that gets base64 encoded, and this can increase the overall size of the email. So when your users send emails that exceed a total size of 15 MB, such emails aren't processed by the server.
SVC_INBOUND_MESSAGE_BATCH_SIZE	Indicates the number of emails that can be processed at a given time. This is the maximum number of emails retrieved by every run of the Retrieve Inbound Email Messages scheduled process. The default value is 10.
SVC_EMAIL_PROCESS_UNKNOWN_CUST	Indicates whether an SR must be created for emails sent by unknown customers.
SVC_ENABLE_INBOUND_EMAIL_ACKNOWLEDGEMENT	Indicates whether an acknowledgment must be sent for an incoming email. If you prefer not to have emails sent when SRs are created, disable this profile option. To disable it, use the Manage Administrator Profile Values task in Setup and Maintenance. Search for the SVC_ENABLE_INBOUND_EMAIL_ACKNOWLEDGEMENT profile option, and then set the profile value to No.
SVC_INBOUND_ACK_EMAIL_LIMIT_PER_USER	Specifies the maximum number of emails to be sent to one user within a specific time interval. This time interval is specified in the SVC_INBOUND_ACK_EMAIL_TIME_INTERVAL profile option. The default value is 3. Note: Let's say you don't want to use this feature, and you want to send an acknowledgment email for every email received from a user. In that case, you can set this profile option to a high value such as 100. It's a good idea to simultaneously set the SVC_INBOUND_ACK_EMAIL_TIME_INTERVAL profile option to a low value such as 5 minutes. This profile option also helps to prevent the creation of infinite email loops.
SVC_INBOUND_ACK_EMAIL_TIME_INTERVAL	Specifies the time interval for which the limit check is applied for the maximum number of emails to be sent to a user. This means that only the maximum number of emails specified in the SVC_INBOUND_ACK_EMAIL_LIMIT_PER_USER profile option can be sent to one user in this time interval. The default value is 60 minutes. Note: Let's say you don't want to use this feature, and you want to send an acknowledgment email for every email received from a user. In that case, you can set this profile option to a low value such as 5 minutes. It's a good idea to simultaneously set the SVC_INBOUND_ACK_EMAIL_LIMIT_PER_USER profile option to a high value such as 100.

Inbound Profile Options	Description
	This profile option also helps to prevent the creation of infinite email loops.
SVC_INBOUND_EMAIL_PATTERN_TO_STOP_ACK	Specifies the email patterns that must be blocked, so that the application stops sending acknowledgment emails to email addresses having those patterns. The default email pattern is postmaster@.*.oraclecloud.com .
ORA_SVC_ENABLE_FAILED_INBOUND_MESSAGE_PROCESSING	Enables the automatic processing of inbound messages that weren't processed earlier because of internal issues. The default value is Yes. To stop the retrieval of unprocessed messages at any time, you can set the value to No.
ORA_SVC_ADD_FIRST_ATTACHMENT_TO_SR_MESSAGE	Enables the agents to see the images or attachments from the first email received on an SR in both the SR header and the SR message. The default value is No.
ORA_SVC_ENABLE_INBOUND_INLINE_ATTACHMENT_CATEGORY	Enables viewing of images and logos inline within the body of the SR message, but the inline images aren't captured as regular SR attachments. The default value is No. When it's set to Yes: - Documents received as attachments to inbound emails are included in the SR attachments. - Images such as logos in the sender's signature and inline images aren't included in the SR attachments. Note: When you set this value to Yes, the ORA_SVC_ADD_FIRST_ATTACHMENT_TO_SR_MESSAGE profile option is automatically set to Yes.
ORA_SVC_INBOUND_MSG_LAST_REPROCESSING_DAYS	Specifies the number of days before the current date when the failed inbound messages are to be reprocessed. The default value is 1.
ORA_SVC_ENABLE_INBOUND_PARTIAL_STATUSES_FOR_REPROCESSING	Enables the reprocessing of unprocessed inbound emails from the past. The default value is No. When it's enabled, reprocessing starts from the last point where the processing stopped earlier.
ORA_SVC_ADD_EMAIL_RECIPIENTS_TO_SR	Enables you to choose whether to add the sender and recipients of inbound emails as an SR contact or SR team member. For a new SR, the recipients in the To or Cc lists are added as SR contacts or SR team members, if their email IDs exist as valid party records in the application. For a reply email to an existing SR, the sender and the contacts or team members in the To or Cc lists are added as SR contacts or SR team members if they don't already exist in the SR. The default value is Yes. When you set the value to No, the sender and recipients of inbound emails aren't added to the SR.

Inbound Profile Options	Description
	Note: The sender of the first email is always added as the primary contact on the SR, irrespective of the value of this profile option.
ORA_SVC_UNREGISTER_INBOUND_ACCESS_POINT	Specifies whether to make the Unregister button available. The default value is No. When it's set to the default value, the Unregister button isn't available. This setting helps to prevent unregistering of access points accidentally. Only when you specifically need to unregister an access point, you can change the value to Yes so that the Unregister button is available. After unregistering the access point, you can set the value of this profile option back to No.
ORA_SVC_INBOUND_REPROCESSING_OFFSET_IN_MINUTES	Used to reprocess inbound emails when it fails to create customer message on the SR. The default value of this profile option is set to 120 (2 hours). After this duration, the reprocessing logic will try to create customer message again on the SR.
SVC_INBOUND_EMAIL_REGEX_TO_EXTRACT_SR_NUMBER	Used to create and send a new message for an existing SR. Note: If the default value can't extract the SR number from the inbound email body, then try adding an extra # to the default value.
ORA_SVC_INBOUND_MSG_EXCHANGE_ONLINE_BATCH_SIZE	Specify the maximum number of inbound messages to read from Microsoft Exchange online in one ESS Job iteration. The default value is 10 The maximum value is: 200
ORA_SVC_INBOUND_MSG_EXCHANGE_ONLINE_FETCH_SIZE_PER_BATCH	Specify the maximum number of inbound messages to be processed from Microsoft Exchange online in one subset of a batch. The default value is 10 The maximum value is: 20
ORA_SVC_INBOUND_MSG_EXCHANGE_ONLINE_DELETE_SIZE	Specify the maximum number of inbound messages to be deleted from Microsoft Exchange online in one subset of a batch. The default value is 10 The maximum value is: 20
SVC_INBOUND_EMAIL_REGEX_TO_EXTRACT	This pattern will be used to extract the SR number from the Inbound email body.
SVC_INBOUND_EMAIL_REGEX_TO_EXTRACT_MSG_NUMBER	This pattern will be used to extract the message number from an inbound email message body.

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

Outbound Profile Options	Description
SVC_OUTBOUND_EMAIL_FROM	Indicates the From email that's used for sending outbound emails to customers. The default value is noreply@oracle.com. You must set the value to the no reply address for your company to prevent auto-replies from creating unintended SRs.

Outbound Profile Options	Description
	This profile option can have the no reply email address with or without the display name. Example without display name: noreply@mycompanydomain.com. Example with display name: Acme Support <noreply@mycompanydomain.com>. If the profile option value doesn't contain a display name, the display name of the email channel is used in the From address. Let's say you set the value of this profile option as your support email ID such as support@mycompanydomain.com. If your From email address isn't displayed in your email received by your customer, then you'd receive a verification email at support@mycompanydomain.com. You must follow the instructions in that email to verify the account. If you don't have access to the inbox for support@mycompanydomain.com, then you must do one of the following: • Contact Oracle support to get your email account verified. • Set the value of this profile option to another valid email address that you can access. Now when you send your first email to a customer, you receive a verification email. You must click the confirmation link in the email to complete the verification.
SVC_SR_FORWARD_TEMPLATE_NAME	Indicates the email template name for SR messages of type Forward.
SVC_SR_RESPONSE_TEMPLATE_NAME	Indicates the email template name for SR messages of type Response.
SVC_SR_SYSTEM_RESPONSE_TEMPLATE_NAME	Indicates the template name for SR messages of type System Response.
ORA_SVC_SR_EMAIL_ATT_SIZE	Indicates the maximum permitted total size in MB of all the attachments in an outbound email that's sent from the Service application. Oracle recommends keeping this value less than or equal to 10 MB. The default value is 10 MB. Here are some key points to note: • The maximum limit for the email size is 15 MB. Emails greater than 15 MB in size aren't processed by the email server. Email size includes email headers, attachments, and the email content. If an email contains non text content or attachments, that gets base64 encoded, and this can increase the overall size of the email. • Use this profile option to restrict the email attachment size such that the total email size doesn't exceed 15 MB.
SVC_ENABLE_ACKNOWLEDGMENT_TO_ALL	Indicates whether an acknowledgment must be sent to users both in the To and Cc lists for every inbound email that's received.
SVC_ENABLE_DEEPLINKS_IN_OUTBOUND_EMAIL	Indicates whether the deep links to KM articles are enabled in outbound emails and email previews. When the deep links are enabled, this helps to translate and populate the complete URL to view the articles.
ORA_SVC_ENABLE_RESOURCE_NAME_USAGE_IN_OUTBOUND	Enables the use of the resource name as the sender name in outbound emails. The default value is No.
ORA_SVC_ENABLE_RESOURCE_EMAIL_USAGE_IN_OUTBOUND	Enables the use of the resource email as the sender email in outbound emails. The default value is No.
ORA_SVC_ENABLE_INLINE_ATTACHMENTS_IN_OUTBOUND_EMAIL	This profile option only impacts inline attachments, not regular attachments.

To configure email profile options:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Inbound Email Profile Options, or Manage Outbound Email Profile Options
2. Click the name of the profile option to be set.
3. In the **Manage Email Profile Options** page, in the **Profile Values** section, click the plus icon to add a value.
4. Click **Save**.

Note: Alternatively, you can click the **Manage Email Profile Options** task to display the Email Profile Options page. On this page, you can view the list of all inbound and outbound email profile options and set their values. You can also search for a specific profile option code using the search bar.

How do I enable the ability to add all contacts and team members or add additional email recipients?

Sometimes while composing an email, an agent might want to add extra email recipients who aren't contacts or resources (team members), or an agent might want to add all the contacts and team members of the service request.

For these options to be available to them, you must enable the `SVC_ENABLE_ADDITIONAL_EMAIL_RECIPIENTS` profile option. The default value of this profile option is No.

To enable the `SVC_ENABLE_ADDITIONAL_EMAIL_RECIPIENTS` profile option:

1. Sign in as a setup user or administrator.
2. Navigate to the Setup and Maintenance work area.
3. Open the Tasks panel tab and click **Search**.
4. Search for and select the **Manage Administrator Profile Values** task.
5. On the Manage Administrator Profile Values page, navigate to the Search region.
6. In the **Profile Option Code** field, type `SVC_ENABLE_ADDITIONAL_EMAIL_RECIPIENTS`.
7. Click **Search**.

The `SVC_ENABLE_ADDITIONAL_EMAIL_RECIPIENTS` profile option is displayed in the Search Results: Profile Options region.

8. In the Profile Option Levels region, set the **Profile Value** for the Site **Profile Level** as **Yes**.
9. Click **Save**.

How do I set the From Name in outbound emails?

When you send emails to your customers, the **From Name** in the outbound email is typically the channel name. But you have the option to set the resource name as the **From Name** in outbound emails.

You can do this by configuring the `ORA_SVC_ENABLE_RESOURCE_NAME_USAGE_IN_OUTBOUND` profile option.

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

1. Ensure that you've configured the SPF policy to enable outbound email so that your emails aren't rejected by your customer's mail server.
2. Sign in as a setup user or administrator.
3. Navigate to the **Setup and Maintenance** work area and open the **Tasks** panel tab.
4. Click **Search** from the list of displayed tasks.
5. Search for and click **Manage Administrator Profile Values**.
6. On the Manage Administrator Profile Values page, navigate to the Search region.
7. In the **Profile Option Code** field, type `ORA_SVC_ENABLE_RESOURCE_NAME_USAGE_IN_OUTBOUND`.
8. Click **Search**.

The `ORA_SVC_ENABLE_RESOURCE_NAME_USAGE_IN_OUTBOUND` profile option is displayed in the Search Results: Profile Options region.

9. In the Profile Option Levels region, set the **Profile Value** for the Site **Profile Level** as **Yes**.
10. Click **Save and Close**.

How do I define email templates?

You can create email templates for the Forward, Response, and System Response messages of a service request (SR).

You can create templates using HTML to send email notifications for an SR using Application Composer.

To define an email template:

1. Sign in to the application as an administrator.
2. Navigate to Application Composer.
3. In the **Application** field, select **CRM Cloud** from the drop-down list.
4. In the **Common Setup** region, click **Email Templates**.
5. On the Email Templates page, click the plus icon to create a new template.
6. In the **Object** field, select **Service Request** from the drop-down list.
7. Specify a name for the template.
8. (Optional) 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, the subject can be `Resolved issue [{Title$}]`.

12. Edit the message HTML as required. Add the `#MessageContent#` tag anywhere in the HTML code. This tag is replaced by the SR message content.
13. (Optional) To enable the display of the email thread, insert the `#PastConversation#` tag in the template.

Add this tag within the email body at the place where you want to insert the previous messages from the email thread. When an agent replies to a customer from the Messages tab in an SR, the email thread is displayed.

14. 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 Fusion Service site>/service/faces/FuseOverview?  
fndGlobalItemId=itemNode_service_service_requests&pSrNumber=<SR Number>.
```

For example, `https://company123.mycompanydomain.com:10616/service/faces/FuseOverview?_afdcGlobalItemId=itemNode_service_service_requests&pSrNumber=SR0000029093`.

15. Click **Save and Close**.

Note: When an SR is created, it might be with or without a queue. So it's possible that an email notification is sent before the SR is assigned to a queue. To avoid a blank field value in the email notification, you must not use the **Queue Name** field in your email template.

How do I change acknowledgment messages for inbound email?

When you receive an email from your customer or an employee to open a service request, an acknowledgment message is sent to them automatically. Based on your company's requirement, you can change the predefined acknowledgment messages provided by the application.

Note: You can't add new tokens to these messages. Only the predefined tokens are used in these messages.

For 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

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

1. In the Setup and Maintenance work area, go to the **Tasks** panel tab.
2. Click **Search** from the list of displayed tasks.
3. In the **Search** window, enter Manage Messages.
4. Click the **Manage Messages** task that's displayed.
5. In the **Manage Messages** window, in the **Message Name** field, enter `%SVC%ACK%`.
6. From the list of acknowledgment messages displayed, select the message that you want to edit and click **Edit**.
7. In the **Edit Message** window, navigate to the **Message Text** region.
8. Edit the **Short Text** and **User Details** the way you want.
9. Click **Save**.
10. To translate the changed messages, click **Translation Editor**.

The **Edit Translations** window displays the list of available languages for translating the messages.

Note: Inbound email can't detect the language. So it always uses the default language for your deployment.

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. Select another row and repeat the same steps 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.

Decide Whether to Add Email Recipients as Service Request Contacts or Team Members

You can decide whether the recipients of an inbound email must be added as service request (SR) contacts or SR team members.

The sender of the first email is always added as the primary contact on the SR. For the follow-up emails, you can choose not to add the sender of the email as an SR contact or SR team member. You can do this by configuring the `ORA_SVC_ADD_EMAIL_RECIPIENTS_TO_SR` profile option.

When you set this profile option to No, the sender of the follow-up emails isn't added to the SR.

Note: The sender of the first email is always added to the SR, irrespective of the value of this profile option.

1. Sign in as an administrator.
2. Navigate to the **Setup and Maintenance** work area.
3. Open the **Tasks** panel tab and click **Search**.
4. Search for and select the **Manage Administrator Profile Values** task.
5. On the Manage Administrator Profile Values page, navigate to the Search region.
6. In the **Profile Option Code** field, type `ORA_SVC_ADD_EMAIL_RECIPIENTS_TO_SR`.
7. Click **Search**.

The `ORA_SVC_ADD_EMAIL_RECIPIENTS_TO_SR` profile option is displayed in the Search Results: Profile Options region.

8. In the Profile Option Levels region, set the **Profile Value** for the Site **Profile Level** as **No**.
9. Click **Save**.

How do I enable deep links to KM articles in outbound emails?

When your service agents reply to customers by email, they might sometimes need to add links to Knowledge Management (KM) articles or other related service requests (SRs). These articles can help resolve the issue faced by the customers in that SR, or they can provide more information.

These internal links to KM articles work in the SR Messages tab. But for these internal links to work in emails sent to customers, they must be translated to external links that open on the customer service application.

Your customers typically have access to one or more customer service applications. These customer service applications provide them access to their SRs and relevant knowledge articles through a web interface. They might use multiple customer service applications for various reasons. For example, customers often use different customer service applications for different business units (BUs).

Here's what you need to do to convert the internal links to external links that open in the customer service application:

- Enable the SVC_ENABLE_DEEPLINKS_IN_OUTBOUND_EMAIL profile option.
- Use the Manage Service Request Dynamic Links task to configure the **External URL** field, as described in the following table.

This table shows how the links work, depending on how the agent links to the article.

Type of link created to the KM article	What to append to the external URL to make the link work	How the link works
Article link format: When the agent uses either the "Insert as link" in RHP or the "Insert Knowledge" button KM application.	? kmContentId={0} &page=shell&shell=know article	The parameter {0} is replaced by the internal document ID of the selected KM article, and a clickable link to the article is inserted in the email message. The title of the KM article is used as the label for the link. The article opens in the customer service application in a separate nested tab.
Knowledge article format: When the agent composes a message and provides a reference to a KM article. For example, if the agent types in the text FAQ1234, this text is linked in the email message.	? kmExternalId={0} &page=shell&shell=know article	The parameter {0} is replaced by the internal document ID of the selected KM article. The article is linked by using the typed text as the label for the link. Example of typed text: FAQ1234. The article opens in the customer service application in a separate nested tab.

Note: The deep links don't work if they contain the # character. Also, special characters such as @, ^, and * can be used only if preceded with the backslash character \. For example, you must use * instead of * in your regular expression. For more information about using special characters in regular expressions, see the open source documentation for regular expressions. After you change the regular expression, remember to use a validator to ensure that it's still working.

To enable deep links to KM articles in outbound emails and email previews to open in the customer service application:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Productivity Tools
 - o Task: Manage Service Request Dynamic Links

Note: Optionally, you can search for the Manage Service Request Dynamic Links task from the Tasks panel tab in the Setup and Maintenance work area.

2. On the Manage Dynamic Links page, repeat steps 3 to 12 for each of the business units (BUs) for your deployment.

That's because you might have different customer service applications for different BUs. If you only have a single BU, you need to do the steps only once.

3. Select the row that has **ARTICLE_LINK** in the **Object Type** field.
4. Click **Add** to add a row.

5. From the **Business Unit Name** drop-down list, select a BU.
6. In the **External URL** field, do the following:
 - a. Enter the external URL provided to you.
 - b. Append the following string at the end, after the slash: `?kmContentId={0}&page=shell&shell=knowledge-article`
This string is explained in the first row of the table.
Example of how the final URL looks: `https://myvbinstance.builder.ocp.oc.com/ic/builder/rt/KMEXTID297/1.0/webApps/dcs/?kmContentId={0}&page=shell&shell=knowledge-article`
7. Click **Save**.
8. Select the row that has **KNOWLEDGE_ARTICLE** in the **Object Type** field.
9. Click **Add** to add a row.
10. From the **Business Unit Name** drop-down list, select the same BU as in step 5.
11. In the **External URL** field, do the following:
 - a. Enter the external URL provided to you.
 - b. Append the following string at the end, after the slash: `?kmExternalId={0}&page=shell&shell=knowledge-article`
This string is explained in the second row of the table.
Example of how the final URL looks: `https://myvbinstance.builder.ocp.oc.com/ic/builder/rt/KMEXTID297/1.0/webApps/dcs/?kmExternalId={0}&page=shell&shell=knowledge-article`
12. Click **Save**.

Add Deep Links to Third Party Applications from Service Request Messages

You can add custom object types and provide dynamic links to third party applications within the service request message content. The user can drill down directly from dynamic links in a service request message into specific records in non Fusion applications.

You can define regular expression patterns that convert to a URL link, when the user types text matching that pattern. For example, when a user types text matching the pattern for a service request reference, the text in the SR message is automatically converted into a link to that SR. This means that a user can, for example, type in **FAQ123**, **SR123**, or **Bug123**, and if the pattern matches, it creates a link in the SR message, to the corresponding FAQ knowledge article, SR, or bug tracking application.

Here's how you configure the third party link for custom objects:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Productivity Tools
 - o Task: Manage Service Request Dynamic Links
2. On the Manage Dynamic Links page, click **Add**.
The **Add** dialog box is displayed.
3. In the **Object Type** field, enter a name for the custom object (for the third party application) you want to create.
4. To automatically generate dynamic link patterns for links that point to the third party application, select the **Generate pattern** check box.

Use this option in case you aren't sure how to reference an object. You can't edit this pattern.

5. Click **OK**.
6. Select the custom object row you added, and click **Add**.
7. From the **Business Unit Name** drop-down list, select a BU.
8. In the **External URL** field, enter the URL that's needed to deep link on the selected custom object.
The external URL should also contain {1}, which is the parameter that's replaced with the text or number adhering to the pattern.
For example, if you add a pattern such as (?i)\b(txt) ([0-9]{3})\b that opens in a URL such as https://abc.com/edit_info?rpt=txt123, then the external URL should be https://abc.com/edit_info?rpt={1}
9. In the **Internal URL** field, enter the internal URL for the third party application (for the selected custom object).
10. Click **Save**.

Note: To delete a custom object type, click **Delete** (the X icon) for the corresponding row. You can't delete the ready to use object types such as service request, article link, and knowledge article.

Here's some more examples that show how to configure other deep links:

- SRs:
 - Pattern: (?i)\b(SR) ([0-9]{10})\b
 - External URL: https://<fusion application URL>/deeplink?objType=SVC_SERVICE_REQUEST&objKey=srNumber%3D{0}&action=EDIT_IN_TAB
 - Internal URL: Not applicable
- HR Help Desk articles:
 - Knowledge_Article (for example, HFAQ123): https://<fusion application URL>/deeplink?objType=CSO_ARTICLE_CONTENT_HCM&objKey=docId%3D{0}%3Blocale%3D{1}&action=EDIT_IN_TAB
 - Article_Link (for example, HFAQ123): https://<fusion application URL>/deeplink?objType=CSO_ARTICLE_CONTENT_HCM&objKey=articleId%3D{0}%3Blocale%3D{1}&action=EDIT_IN_TAB

Access Point Setup for Inbound Emails

Overview of Access Point Setup for Inbound Emails

When your customers send emails to your support channel, the User Messaging Service (UMS) reads those emails from your inbox and delivers them to registered access points.

Access points are registered clients that are authorized to read emails from your inbox. The application then reads those emails from the access points.

For inbound email to be processed successfully, it's important to register the correct access points with the UMS. You can use the Access Point Setup region of the Inbound Email Configuration and Validation page to register access points. When you register an access point, emails are permitted to be processed further. For example, to create a service request. When you unregister an access point, further processing of emails is stopped, and the emails in your inbox are also lost without being processed.

Related Topics

- [How do I validate my inbound and outbound email setup?](#)
- [How do I configure an email channel?](#)

How do I validate my inbound and outbound email setup?

You can verify your inbound and outbound email configurations by using the Inbound Email Configuration and Validation page.

On this page, you can proactively take corrective actions to fix any wrong or missing configuration. The validations on this page save you the trouble of raising a support ticket or checking the logs.

Use the Inbound Email Configuration and Validation page to do the following validations:

- Inbound email channel profile options
- Channel setup
- Access points for inbound email
- Outbound email flow
- Inbound email flow

To verify your email channel configurations:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Communication Channels
 - Task: Manage Email Configuration, Registration, and Validation

The Inbound Email Configuration and Validation page is displayed.

2. In the Email Configuration region, click **Validate**.

The following validations happen in this region:

- **Channel Setup:** Checks whether you've done the following:
 - Configured at least one non predefined channel.
 - Specified a Business Unit for every channel.
- **Profile Option:** Reads the SVC_INBOUND_EMAIL_ADDRESSES profile option for email addresses and displays an error message if email addresses aren't defined.
- **Access Point Setup:** Checks whether the access points are registered appropriately. It's important to register the correct access points.

For each setup or configuration that's valid, a tick mark is displayed.

If the configuration is invalid or the setup isn't done correctly, a cross mark is displayed. A suitable message is displayed, describing the corrective action that you must take.

3. In the Email Flow region, in the **Recipient** field, enter your company's test mail ID.

The email flow for the outbound and inbound emails is validated here.

4. Click **Validate**.

The following validations happen in this region:

- o Outbound Email Validation:

An outbound email is sent to the specified email address and it's verified whether the message is delivered successfully.

- If the host, port, or password are incorrect, a suitable error message is displayed and the status is displayed as DELIVERY_TO_GATEWAY_FAILURE.
- If the email is delivered successfully, the status code is displayed as DELIVERY_TO_GATEWAY_SUCCESS.

- o Inbound Email Validation:

The following validations happen:

- i. Test emails are sent to all the email addresses specified in the SVC_INBOUND_EMAIL_ADDRESSES profile option and the delivery status is verified for each message.
- ii. If an email address isn't registered as an access point, then you can't receive emails from that email address. The status appears as DELIVERY_TO_CLIENT_FAILURE. An error message is displayed, explaining the suitable action to be taken.
- iii. For each email address specified in the SVC_INBOUND_EMAIL_ADDRESSES profile option, a success or failure message is displayed along with the status.

The status is displayed as:

- o DELIVERY_TO_CLIENT_SUCCESS: When the inbound email is received successfully by the access point.
- o DELIVERY_TO_CLIENT_PENDING: When the message is received but not yet processed. You must verify whether the inbound poll job is scheduled.
- o DELIVERY_TO_CLIENT_FAILURE: When the access point isn't registered.

5. Click **Reset** to clear the **Recipient** field and all the messages related to the recipient.

How do I register or unregister access points for inbound email?

For your inbound emails to be processed successfully, you must register the email addresses in the SVC_INBOUND_EMAIL_ADDRESSES profile option as access points. If your inbound email address changes for some reason, then you must register the new inbound email address as an access point.

Sometimes, you might must unregister a wrong email address that was registered by mistake. The **ORA_SVC_UNREGISTER_INBOUND_ACCESS_POINT** profile option controls the ability to unregister access points. Its default value is No. When it's set to the default value, the Unregister button for an access point isn't available. This setting helps to prevent unregistering of access points accidentally. Only when you specifically need to unregister an access point, you can change the value to Yes so that the **Unregister** button is available. After unregistering the access point, you can set the value of this profile option back to No.

CAUTION: When you register or unregister access points, they don't reflect in all the server instances, and some inbound emails might not be received. To prevent this from happening, you can do one of the following:

- Restart the servers to reflect the registering or unregistering of access points. You must contact Oracle Support for restarting the servers.
- Configure the **Retrieve Inbound E-mail Messages** job to run at a suitable frequency. See *How do I configure a job to process inbound emails?*

To register the access points for inbound email:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Functional Area: Communication Channels
 - Task: Manage Email Configuration, Registration, and Validation
2. On the Inbound Email Configuration and Validation page, in the Email Configuration region, click **Validate**.

The access point setup is validated in this region.

The following email addresses are displayed in the **Access Point Value** column of the Access Point Setup region:

- Email addresses specified in the SVC_INBOUND_EMAIL_ADDRESSES profile option.

Suppose one or more of these email addresses aren't registered as an access point with the UMS. In such cases, a **Register** button appears in the **Action** column for these email addresses.

- Email addresses that are registered as access points in the UMS.

All email addresses that are registered as access points with the UMS are displayed in this section. An **Unregister** button appears in the **Action** column for each email address.

Note: Don't click **Register** and **Unregister** at this stage. You must complete the remaining steps.

3. Verify whether the email addresses specified in the SVC_INBOUND_EMAIL_ADDRESSES profile option are correct.
4. If the email addresses in the SVC_INBOUND_EMAIL_ADDRESSES profile option aren't correct, update them to show the correct values.
5. Refresh the Inbound Email Validation and Configuration page.
6. If you still see a **Register** button, click the button to register the email address with the UMS.
7. If you still see an **Unregister** button, click the button, because the email address is wrong.

The wrong email address is unregistered, and is no longer an access point with the UMS.

Note:

- When you unregister an email address, all inbound emails to that email address are lost.
- If the **Unregister** button appears grayed out and you can't click it, set the ORA_SVC_UNREGISTER_INBOUND_ACCESS_POINT profile option to Yes as explained earlier in this topic. Now you can click the **Unregister** button.

Overview of Times Displayed for the Access Points

This table describes the meaning of the different times displayed for each email address.

Column Name	Description
Last Read Time	When a time is displayed in this column, and the other two columns are blank, this indicates the following: An email address is configured with the UMS server, but the access point isn't registered. So, the UMS reads the email from the inbox but isn't able to deliver it to the access point, because an access point doesn't exist. Maps to the DELIVERY_TO_CLIENT_FAILURE status of the UMS server.
Last Successful Read Time	When the Last Successful Process Time is blank, this indicates that the access point is registered, but the scheduled inbound poll job hasn't run. The UMS reads the email from the inbox and delivers it to the access point, but the scheduled job hasn't run. Maps to the DELIVERY_TO_CLIENT_PENDING status of the UMS server.
Last Successful Process Time	Indicates the last time when the email was read by the scheduled inbound poll job. Maps to the DELIVERY_TO_CLIENT_SUCCESS status of the UMS server.

How You Interpret the Times Displayed for the Access Points

This table shows some examples of times displayed for each sample email address in the Access Point Setup region.

No.	Access Point	Last Read Time	Last Successful Read Time	Last Successful Process Time
1.	inbound_email1@oracle.com	7/17/2018 3:00 a.m.	7/17/2018 3:06 a.m.	7/17/2018 3:06 a.m.
2.	inbound_email2@oracle.com	7/19/2018 7:30 a.m.		
3.	inbound_email3@oracle.com	7/19/2018 7:36 a.m.	7/18/2018 7:36 a.m.	7/18/2018 7:36 a.m.
4.	inbound_email4@oracle.com	7/19/2018 7:36 a.m.	7/19/2018 7:36 a.m.	

No.	Access Point	Last Read Time	Last Successful Read Time	Last Successful Process Time
5.	<code>inbound_email15@oracle.com</code>			

Here's how you interpret the above examples of times displayed for the access points in the Access Point Setup region:

Possible scenarios	What this means	What you need to do
<code>inbound_email11@oracle.com</code> in the previous table: - Last Read Time has a value. - Last Successful Read Time has a value. - Last Successful Process Time has a value. - One of the following conditions is satisfied: - All the values are synchronized. - The Last Successful Read Time and Last Successful process time values aren't earlier than the Last Read Time.	The process works correctly without any issues.	You don't need to do anything.
<code>inbound_email12@oracle.com</code> and <code>inbound_email13@oracle.com</code> in the previous table: One of the following conditions is satisfied: - Last Read Time is later than the Last Successful Read Time. - Last Read Time has a value, but Last Successful Read Time and Last Successful Process Time are blank.	An email is read by the UMS server, but the email isn't delivered to the access point. Note: After you register the access point, emails are successfully delivered to the access point.	Do the following: - Starting from the Last Successful Process Time, verify till the current time and see if any Last Successful Read Time exists. - For <code>inbound_email12@oracle.com</code>, starting from about 30 to 40 hours earlier than the Last Read Time, verify whether anything is wrong with respect to the access point. For example, the initial setup may be wrong. - For <code>inbound_email13@oracle.com</code>, a Last Successful Read Time is displayed, and it's later than the Last Successful Process Time. This implies that a mail has been read recently, but hasn't been processed successfully. - See if any records have the DELIVERY_TO_CLIENT_FAILURE or DELIVERY_TO_CLIENT_PENDING status. For more information about searching for records, see "Troubleshoot Access Point Issues". - Register the access point.
<code>inbound_email14@oracle.com</code> in the previous table: Last Successful Process Time is blank.	The scheduled inbound poll job has never received an email.	Do the following: Check whether the inbound poll job called Retrieve Inbound Email Messages is scheduled or if an error is occurring.

Possible scenarios	What this means	What you need to do
		2. If the job isn't scheduled, then schedule a new job by going to Scheduled Processes from the menu in the application.
<code>inbound_email15@oracle.com</code> in the previous table: All the columns are blank.	The UMS is unable to read emails from your inbox either because the UMS setup isn't done or the UMS configuration is wrong.	Do the following: 1. Verify whether the UMS setup is completed. Complete the setup if required. 2. Verify whether the UMS configuration is correct. Correct the configuration if required.

Summary of Tips to Interpret the Times Displayed for Access Points

Here's a brief summary of tips to interpret the different times displayed for each access point and the status of the records in the UMS:

How the Times Are Displayed for an Access Point	What this indicates
One of the following conditions is true: - Only the Last Read Time has a value and the other columns are blank. - The values in other columns are earlier than the value in Last Read Time.	The access point is probably not registered and you may be losing emails.
The Last Successful Process Time is blank or earlier than the Last Successful Read time.	The emails aren't being processed by the inbound scheduled job.
The difference between the times listed in all the three columns is less.	The access point is working fine.

Troubleshoot Access Point Issues

If some emails aren't processed for a particular access point, a particular sender, or within a time range, you can search for those specific records. The records retrieved from the UMS are sorted by date and they show the status in the UMS.

Use the search icon in the Access Point Setup region to search for specific records in the UMS based on the following: From Time, To Time, Sender, or Receiver. You can filter the retrieved records by the access points that are registered, by sender, or by receiver.

To troubleshoot access point issues:

1. On the Inbound Email Configuration and Validation page, navigate to the Access Point Setup region.
2. Click the **Search** icon.
3. Depending on your requirement, enter the details in one or more of the following fields in the **Search Message** window: **From Time**, **To Time**, **Sender**, and **Receiver**.
4. Click **Search**.

The Search Results are displayed. The records are sorted in the descending order of time, with the latest ones on top. The following details are displayed for each record that is retrieved:

- Message ID
 - Recipient
 - Date
 - Status
5. View the details and take suitable action.

Example of a Scenario Where Inbound Email Fails

This is an example of a case where inbound email may fail.

Let's say that your support mailbox is support@mycompanydomain.com and you have set up a forwarding rule. When your customer sends an email to support@mycompanydomain.com, the mail is forwarded to mycompanydomain.extservice@oraclecloud.com. The email address mycompanydomain.extservice@oraclecloud.com is registered as an access point. So, the inbound emails work fine.

However, suppose you upgrade your installation and a password error occurs after the upgrade. If someone changes the email address in the SVC_INBOUND_EMAIL_ADDRESSES profile option to mycompanydomain2.extservice@oraclecloud.com, then your inbound emails stop working for the following reasons:

- The updated email address in the SVC_INBOUND_EMAIL_ADDRESSES profile option isn't registered as an access point with the UMS.
- The email address in the forwarding rule doesn't match the updated email address in the SVC_INBOUND_EMAIL_ADDRESSES profile option.

When the email address in the SVC_INBOUND_EMAIL_ADDRESSES profile option is changed, the following changes are reflected in the Access Point Setup section of the Inbound Email Configuration and Validation page:

- For the new email address that's added to the SVC_INBOUND_EMAIL_ADDRESSES profile option, a **Register** button appears.
- For the old email address in the SVC_INBOUND_EMAIL_ADDRESSES profile option, an **Unregister** button appears.

Note: It's important that you first correct the email addresses and refresh the Inbound Email Configuration and Validation page. And only after that, you must click **Register** and **Unregister** as required.

Job to Process Inbound Emails

How do I configure a job to process inbound emails?

You can configure a new job to retrieve emails at regular intervals, based on the specified frequency.

Note: Inbound email is an asynchronous channel. When you have a high volume of emails, the emails are queued up and processed. Inbound email processing can take anywhere from 10 minutes up to 1 hour from the time the user sends an email depending on your inbound email volume and the frequency at which you've scheduled ESS job to process inbound emails.

To configure a job to process inbound emails:

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 Email Messages** option from the **Name** drop-down list.
5. Click **OK**.
6. In the **Process Details** dialog box, 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**.

Note: You must ensure that you always select a frequency, whether you're using your test or your production environment. You can select a higher frequency if required. If you run a job manually without setting a frequency, the process might not retrieve all the emails. When you schedule the job to run periodically, it pulls all the emails from all the servers. After scheduling the ESS job your application may take few iterations of the job to refresh the caches that are required for the successful execution of inbound email processing. As a result, email processing might be delayed until the required caches are fully refreshed.

9. Click **Submit**.

A job is scheduled to process inbound emails.

Note: Inbound emails work properly only for SRs whose reference numbers have the following characters: [A-Z], [a-z], [0-9], and the special characters hyphen (-), colon (:), period (.), comma (,), or underscore (_). So you can use only one of these characters as the prefix for an SR reference number. You can't use any other special character in the prefix apart from the ones mentioned.

Display Email Thread in Preview and Outbound Email

How do I display email thread of past conversations?

When an agent responds to a customer's email from the Messages tab in a service request (SR), it's a good idea to display the email thread with the past conversations. This way, the customer easily understands the context.

This is also useful for agents as the email thread is included when the email is previewed. Past conversations can be viewed in the thread latest replies can be written as required.

To enable the display of the email thread, you need to complete a few simple processes:

- Configure a few profile options
- Change email templates to display the email thread

Related Topics

- [How do I configure profile options to display an email thread?](#)
- [How do I change email templates to display the email thread?](#)
- [How do I define email templates?](#)

How do I configure profile options to display an email thread?

When an agent responds to a customer's email from the Messages tab in a service request, it's good to display the email thread. To enable the display of the email thread, you must configure the following profile options.

Profile Option	Description
SVC_EMAIL_ENABLE_PAST_CONVERSATIONS	Indicates whether to display past conversations. To enable past conversations, set the value to Yes. The default value is No, so it's disabled by default.
SVC_EMAIL_NO_OF_PAST_CONVERSATIONS	Indicates the number of past conversations to be included in the email thread. The default value is 1. You can specify a value from 1 to 10. If you set the value to 0, no past conversations are included. And even if you set a value greater than 10, only a maximum of 10 conversations are displayed.
SVC_EMAIL_PAST_CONVERSATION_START_MARKER	Indicates the start marker for each conversation in the email thread. This value is fixed and is displayed in English by default. To translate and display this value in the language of your choice, you must set the required locale in the ORA_SVC_EMAIL_DEFAULT_LANGUAGE profile option.

Profile Option	Description
SVC_EMAIL_PAST_CONVERSATION_END_MARKER	Indicates the end marker for each conversation in the email thread. This value is fixed and is displayed in English by default. To translate and display this value in the language of your choice, you must set the required locale in the ORA_SVC_EMAIL_DEFAULT_LANGUAGE profile option.
ORA_SVC_EMAIL_DEFAULT_LANGUAGE	Indicates the default language to format outbound emails and parse inbound emails. The default value is blank. When it's blank, the existing values for the start and end markers are displayed for the past conversations. For details about these markers, see the preceding two rows in this table. You can specify the locale as a value for this profile option. For example: en for English, fr for French, and pt for Portuguese. When you specify the locale, the translated values for the following start and end conversation markers are displayed in the specified language: - SVC_EMAIL_PAST_CONVERSATION_START_MARKER: [##Send your response as a reply to the above mail##] - SVC_EMAIL_PAST_CONVERSATION_END_MARKER: [##End of conversation##]
SVC_EMAIL_PAST_CONVERSATION_MSGTYPES	Indicates the types of messages to include in the email thread. You must specify the codes for the message types, separated by commas. For example: ORA_SVC_INTERNAL_NOTE, ORA_SVC_RESPONSE, ORA_SVC_SYSTEM_NOTE, ORA_SVC_CUSTOMER_ENTRY, ORA_SVC_FORWARD, ORA_SVC_SYSTEM_RESPONSE. You can't specify the meaning or the display name of the message type. The default value is ORA_SVC_CUSTOMER_ENTRY. Note: To add codes for other message types, see the "Search for Lookup Codes of Message Types" section at the end of this topic.
ORA_SVC_EMAIL_PAST_CONVERSATION_SHOW_MSG_TYPES	Indicates whether to include the message type in past conversations. The default value is Y. The message type is translated based on the signed-in user's locale. But the default language might be different. Let's say the signed-in agent's locale is Japanese and the default language is Chinese. So while everything gets translated to Chinese, only the message types appear in Japanese. If your customers contact you about this issue, you can set the value to No. The message type is no longer included in past conversations.
ORA_SVC_EMAIL_DEFAULT_PAST_CONVERSATIONS_IN_OUTBOUND_MSGS	Indicates whether to enable past conversations by default in outbound messages.
ORA_SVC_EMAIL_PAST_CONVERSATION_CHAN_TYPES	Includes messages from only one or more specific channels in past conversations. Set the value to a comma-separated list of those specific channel types that you want.

To configure the profile options for displaying the email thread:

1. Navigate to the Setup and Maintenance work area and open the Tasks panel tab.

2. Click **Search** from the list of displayed tasks.
3. In the **Search** field, type **Manage Admin.**
4. From the displayed list of tasks, click **Manage Administrator Profile Values**.
5. In the Manage Administrator Profile Values page, navigate to the **Search: Profile Option** region.
6. In the Profile Option Code field, type %.
7. Select **%PAST_CONVERSATION%** from the options displayed.
8. Click **Search**.

The Search Results: Profile Options region displays the list of profile options related to past conversations.

9. Select the profile options one by one.
10. In the Profile Values region, specify the values in the **Profile Value** column for the **Site** Profile Level as described in the table.
11. Repeat the previous step for each profile option in the search results.
12. Click **Save**.

Search for Lookup Codes of Message Types

Follow these steps to search for the lookup codes of message types that you want to add in the SVC_EMAIL_PAST_CONVERSATION_MSG_TYPES profile option. You can use the same steps to search for the lookup codes of message types available in the Compose menu in the Messages sub tab of an SR.

1. Sign in as an 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. Search for and select **Manage Standard Lookups**.
5. In the Manage Standard Lookups page, in the **Lookup Type** field, type **ORA_SVC_MESSAGE_TYPE_CD**.
6. Click **Search**.
7. From the **Lookup Code** column in the ORA_SVC_MESSAGE_TYPE_CD: Lookup Codes region of the page, copy the lookup codes that you want to specify in the SVC_EMAIL_PAST_CONVERSATION_MSG_TYPES profile option.

Note:

- The lookup codes ORA_SVC_CHAT_TRANSCRIPT and ORA_SVC_WRAP_UP aren't applicable to emails.
- If you disable any of the following lookup codes, the corresponding message types disappear from the Compose menu on the Messages sub tab of an SR: ORA_SVC_INTERNAL_NOTE, ORA_SVC_RESPONSE, ORA_SVC_CUSTOMER_ENTRY, and ORA_SVC_FORWARD.

How do I change email templates to display the email thread?

When an agent replies to a customer from the Messages tab in an SR, it's good to display the email thread. This way, the customer clearly understands the context.

For this to happen, you must insert the `#PastConversation#` tag into the email templates for SRs. This tag indicates the place within the email layout where you want to insert the previous messages from the email thread.

When the `#PastConversation#` tag is detected, the email-sending process automatically inserts the following details of previous messages from the SR:

- Type of message
- Sender
- To/Cc
- Received date and time of the message for customer entries
- Sent date and time of the message for agent responses
- Creation date and time for other messages
- Message content

When agents view an email, the email thread is included in the preview. This helps them to verify the flow and make changes as required, so the message is clearly understood.

To change an email template to display the email thread:

1. Sign in to the application as an administrator.
2. Navigate to Application Composer.
3. In the **Application** field, select **CRM Cloud** from the drop-down list.
4. Click **Email Templates** from the Common Setup region or the Overview page.
5. From the **Object** drop-down list in the Email Templates page, select **Service Request**.
6. From the **Active** drop-down list, select **Yes**.
7. Click **Search**.
8. From the list of templates displayed, select the template that you want to change.
9. Click **Edit**.
10. Navigate to the Email Body region of the email template.
11. Insert the `#PastConversation#` tag at the place within the email body where you want the email thread to be displayed.
12. Click **Save and Close**.

Related Topics

- [How do I display email thread of past conversations?](#)
- [How do I configure profile options to display an email thread?](#)
- [How do I define email templates?](#)

Configure Email Templates to Control When to Include Past Conversations

When past conversations are enabled by setting the `SVC_EMAIL_ENABLE_PAST_CONVERSATIONS` profile option to Y, they're included in the SR message. But sometimes, you may not want to include past conversations in the SR message.

If the past conversations are enabled and you have used an email template, you can still control whether to include past conversations in an outbound email message. The `ORA_SVC_EMAIL_DEFAULT_PAST_CONVERSATIONS_IN_OUTBOUND_MSGS` profile option lets you decide whether you want to enable past conversations by default in outbound messages. This gives you more flexibility.

Let's consider some examples.

- **Past conversations aren't required:** When an agent is helping your customer to fix an issue, the customer may not require the thread of past conversations in every email related to this issue. So you can set the

ORA_SVC_EMAIL_DEFAULT_PAST_CONVERSATIONS_IN_OUTBOUND_MSGS profile to No and remove the `#PastConversation#` tag from the Customer Response email template. The past conversation isn't included in the email sent to the customer.

- **Past conversations are required:** Let's suppose the agent needs to take the help of another agent to resolve the customer issue. So he must share the whole past conversation thread with the agent who's helping him resolve the issue. In this case, you can set the ORA_SVC_EMAIL_DEFAULT_PAST_CONVERSATIONS_IN_OUTBOUND_MSGS profile to No and add the `#PastConversation#` tag to the Forward email template.

Note: You can configure this profile option by using the Manage Administrator Profile Values task. For more detailed steps, see Configure Profile Options to Display Email Thread.

To include past conversations only in selected emails of your choice, perform the following steps:

- Set the SVC_EMAIL_ENABLE_PAST_CONVERSATIONS profile option to Yes.
- Set the ORA_SVC_EMAIL_DEFAULT_PAST_CONVERSATIONS_IN_OUTBOUND_MSGS profile option to No.
- Add the `#PastConversation#` tag in your email templates for messages where you want to include past conversations.

This table explains how the ORA_SVC_EMAIL_DEFAULT_PAST_CONVERSATIONS_IN_OUTBOUND_MSGS profile option and the `#PastConversation#` tag help you to include or exclude past conversations.

Value of ORA_SVC_EMAIL_DEFAULT_PAST_CONVERSATIONS_IN_OUTBOUND_MSGS Profile Option	#PastConversation# tag present?	Past Conversations included in outbound message?
Yes	Yes	Included
Yes	No	Included
No	Yes	Included
No	No	Not Included

Include Messages of Only Some Channel Types in Past Conversations

On the SRs, there may be various messages from different channels: your customer may have called, visited a retail store, or sent an email. These all become SR messages, but you may want the past conversations to include messages of only some channel types.

You can select those specific channel types by configuring this profile option.

ORA_SVC_EMAIL_PAST_CONVERSATION_CHAN_TYPES: Includes messages from only one or more specific channels in past conversations. Set the value to a comma-separated list of those specific channel types that you want.

Note: You can configure this profile option by using the Manage Administrator Profile Values task. For more detailed steps, see Configure Profile Options to Display Email Thread.

This table shows a sample range of values for this profile option, along with the result.

Profile option value	Messages from the following channel types are included in the past conversations:
blank	All channel types such as Email, Web, and Chat
ORA_SVC_EMAIL	Email only
ORA_SVC_EMAIL, ORA_SVC_CHAT	Email and Chat only
ORA_SVC_EMAIL, ORA_SVC_WEB	Email and Web only

Purge Inbound Email Messages

Purge Inbound Email Messages

As an administrator, you must periodically purge the older inbound email messages after the SRs are created successfully.

Again, you may sometimes have unprocessed emails for which the corresponding SRs couldn't be created. They're typically retained so that the source of messages isn't lost. You must also review them periodically and take suitable corrective action. Once that's done, you can purge those emails.

Note: You can use Oracle BI Publisher to create periodic reports for unprocessed emails.

To purge the unprocessed inbound messages periodically, you have to complete a few simple processes:

- Configure profile options to purge inbound email messages
- Schedule a job to purge inbound email messages

Configure Profile Options to Purge Inbound Email Messages

You must configure the following profile options for purging inbound email messages.

Profile Option	Description
SVC_INBOUND_MSG_RETENTION_DAYS	Indicates the number of retention days for inbound email messages. After the specified number of retention days, the successfully processed email messages are deleted from the inbound message table, inbound message parts table, and inbound message batch table. The default value is 30.
SVC_INBOUND_ENABLE_FAILED_MESSAGE_PURGE	Indicates whether to enable purging of inbound email messages that fail or aren't processed successfully. The default value is No. When the value is No, inbound email messages that fail or aren't processed successfully, are retained. After you review such email messages and take corrective action, you can set the value to Yes. When the value is set to Yes, the purge job deletes the corresponding records from the following tables: • Inbound message table • Inbound message parts table • Inbound message batch table

1. Navigate to the Setup and Maintenance work area and open the **Tasks** panel tab.
2. Click **Search** from the list of displayed tasks.
3. In the **Search** field, type **Manage Administrator**.
4. From the displayed list of tasks, click **Manage Administrator Profile Values**.
5. In the Manage Administrator Profile Values page, navigate to the Search: Profile Option region.
6. In the **Profile Option Code** field, type `SVC_INBOUND_ %`.
7. Click **Search**.

The Search Results: Profile Options region displays a list of profile options.

8. Select **SVC_INBOUND_MSG_RETENTION_DAYS**.
9. In the Profile Values region, specify the values in the **Profile Value** column for the Site **Profile Level** as described in the table.
10. Select **SVC_INBOUND_ENABLE_FAILED_MESSAGE_PURGE** and repeat the previous step.
11. Click **Save**.

Schedule a Job to Purge Inbound Email Messages

You can schedule a job that purges the successfully processed email messages from the inbound message database tables, based on the retention policy.

The retention days are set in the SVC_INBOUND_MSG_RETENTION_DAYS profile option. If you also want this job to purge inbound email messages that fail or aren't processed successfully, you must set the value of the SVC_INBOUND_ENABLE_FAILED_MESSAGE_PURGE profile option to Yes. For more information, see "Configure Profile Options to Purge Inbound Messages".

This job runs periodically to purge inbound email and related records from the following database tables, based on the above two profile options:

- Inbound message
- Inbound message parts

- Inbound message batch

You can schedule the job to run once daily. Or, depending on your company's requirements, you can decide another frequency for the job.

To set up a scheduled process to purge data from the three inbound message tables:

1. In the Navigator for your service application, click **Scheduled Processes**.
2. In the Overview page, click **Schedule New Process**.
3. Select **Job** as the **Type** option.
4. In the **Name** drop-down list, click **Search**.
5. Search for and select **Purge Inbound Messages**.
6. Click **OK**.
7. In the **Process Details** dialog box, click **Advanced**.
8. On the Schedule tab, select **Using a schedule** as the **Run** option.
9. Schedule the job to execute on a recurring basis by specifying the frequency and other details as required.
10. Click **Submit**.

Send Bulk Emails

Optimize Sending Emails in Bulk

You may sometimes have situations where a major incident occurs, such as the network going down in the West Coast of the US. So a huge number of your customers would report issues triggered by this incident.

The root cause for all these issues would be the same incident. In such cases, you can compose multiple outbound messages or SR messages in bulk mode. You can then send out these bulk messages in batches without affecting the normal email communication with customers. Using groovy scripts or REST APIs, application developers can send outbound emails in bulk.

To optimize sending emails in bulk using groovy scripts:

In Application Composer, create an After Create trigger on the service request Message object and add the following groovy script in the **Edit Script** field:

Trigger: Create

```
if (ChannelTypeCd == "ORA_SVC_EMAIL") {  
  
    setAttribute('ProcessingStatusCd', "ORA_SVC_BATCH_READY");  
  
}
```

To optimize sending emails in bulk using REST APIs:

1. Go to Oracle Help Center.
2. Drill down to your apps service area of interest.
3. Go to the APIs and Schema section.
4. Review the REST service definition in the REST API guides available here.

Note: If you're new to Oracle's REST services, you may want to begin with the Quick Start section.

Here are some key pointers:

- If you're using REST APIs to create outbound messages and want to send them out in batch: You must set the value of the `deliveryTypeCd` attribute to `ORA_SVC_BATCH`.
- If you're using REST APIs to create outbound SR messages and want to send them out in batch: You must set the value of the `processingStatusCd` attribute to `ORA_SVC_BATCH_READY`.
- After you create all the outbound messages or SR messages, you can run the Send Outbound Messages scheduled process to send these messages in batch.

Detect Unprocessed Emails

Detect Your Unprocessed Inbound Emails

The records that are read from the email server are stored as records in the database table of your service application. The Detect Unprocessed Emails scheduled process checks the status of the records in the email server from the time that it starts.

It compares the data between the email server and the database tables of your service application. It then identifies the unprocessed records. Every time it runs, it starts checking from the first remaining unprocessed record.

Some emails may be present in the email server, but they may not have reached your service application because the application is down or because of some other issue. Every time the scheduled process runs, it checks the status of these pending emails, starting from the first remaining pending email. After a certain specified time limit, if a pending email has still not reached your service application, it will remain unprocessed.

Note: You can also use BIP reports to check whether you have any unprocessed emails.

Configure Profile Options to Detect Unprocessed Emails

You must configure the following profile option for detecting unprocessed emails.

Profile Option	Description
ORA_SVC_MAX_DURATION_FOR_PENDING_EMAIL	Indicates the maximum duration in hours for which an inbound email can remain in the pending status. The default value is 6 hours. The Detect Unprocessed Emails scheduled process keeps on checking the pending inbound email records for this duration. After this duration, the records that still remain pending are classified as a failure.

To configure the profile option for detecting the unprocessed inbound emails:

1. Navigate to the **Setup and Maintenance** work area and open the **Tasks** panel tab.
2. Search for and click the **Manage Administrator Profile Values** task.
3. On the Manage Administrator Profile Values page, navigate to the Search: Profile Option region.

4. Search for and select the profile option in the table.
5. In the Profile Values region, specify the values in the **Profile Value** column for the Site **Profile Level** as described in the table.
6. Click **Save**.

FAQs for Service Email

Frequently Asked Questions (FAQs) for Service Email

This section contains some frequently asked questions (FAQs) when setting up and maintaining service email. Scroll or search to find answers to common questions.

What happens when a customer sends a service email?

If a customer sends an email, then the email ID 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 service request (SR) is created.

If a match to the email ID isn't 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 SVC_EMAIL_PROCESS_UNKNOWN_CUST 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 SVC_EMAIL_PROCESS_UNKNOWN_CUST 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 Messages tab.

How can I detect and prevent email loops?

You can prevent your service application from creating additional service requests (SRs) when there's an out-of-office reply from one of the recipients of the email.

For outbound SR emails, the recommended best practice is to have your administrator set the From address to a no-reply option, such as `noreply@mycompanydomain.com`. This way, even if one or more recipients have their out-of-office notification turned on, the out-of-office reply isn't received by your service application. So a new SR isn't created.

But let's say your administrator sets the From address to your Support email account. This could cause an infinite loop of emails and new SRs. That's because a new SR sends an automatic acknowledgment email, which would trigger an out-of-office reply. This reply would in turn trigger the creation of another new SR, and so on.

Your business requirements may not permit the use of a no-reply email address, and you may use your Support email account instead. To accommodate this requirement, the following profile options are provided to detect and prevent the creation of infinite email loops:

- SVC_INBOUND_ACK_EMAIL_LIMIT_PER_USER
- SVC_INBOUND_ACK_EMAIL_TIME_INTERVAL
- SVC_INBOUND_EMAIL_PATTERN_TO_STOP_ACK

You can configure these profile options and set the values based on your company's requirement. For more information, see [Configure Profile Options for Inbound and Outbound Email](#).

Related Topics

- [How do I configure profile options for inbound and outbound email?](#)

Why do some emails show as delivered in the configuration menu, but the SRs aren't created?

Sometimes, you may see that some emails appear as delivered when you click the Manage Email Configuration, Registration, and Validation task, but you don't see the service requests (SRs).

These emails also don't appear on the inbound email reports. This is especially true if you implement HR Help Desk.

Let's see why this happens. Suppose you're implementing HCM or HR Help Desk for your deployment. Then you must create two email channels: one with the CRM application stripe and the other with the HCM or HR Help Desk application stripe. But let's say you forget to create the HCM or HR Help Desk email channel. Then the SRs created from the emails are set with the CRM application stripe. So the HCM or HR Help Desk SRs aren't visible, and only the CRM SRs are visible.

To make the HCM or HR Help Desk SRs visible:

1. Create the HCM or HR Help Desk channel.
2. Run an ODI import process to update the stripe code for such SRs from CRM to HCM or HR Help Desk.

After the ODI import, these HCM or HR Help Desk SRs are visible.

Related Topics

- [How do I configure an email channel?](#)

Can I use the problem description of an SR as the subject in an outbound email?

You must not use the problem description of a service request (SR) as the subject in an outbound email.

Problem descriptions for SRs can be very long, sometimes up to 1000 characters. So they can span across multiple lines and also include the ENTER key. That's why it's not valid to use them as the subject in the email template.

Why is the channel ID set to the default email channel for some of my inbound emails?

Let's say your customer sends an email to your support email ID such as `support@mycompanydomain.com`, and it's redirected to the Oracle mailbox. Here's what happens:

- If either the **To** or **CC** address specified by your customer matches the account name of any channel defined in the application, that channel ID is set on the service request (SR).
- If neither the **To** or **CC** address specified by your customer doesn't match the account name of any channel defined in the application, then the channel ID on the SR is set to the default channel, which is Email.

How can I prevent a new SR from being created when a customer replies to an email notification that's set up using Groovy scripts or object workflow?

For all outbound emails generated by your service application, the message ID is automatically added at the end of the email in the format `{##MessageId##}`. When someone replies to such an outbound email or forwards it, the application uses these values to identify the parent SR.

But let's suppose you've created a Groovy script or an object workflow to send an email notification when an SR is resolved. And this notification doesn't have the `{##MessageId##}` or `{##SrNumber##}` tags at the end of the message. So when your customer's SR is resolved, they receive a notification without the identifying tags. Now let's say the issue recurs for the customer. Instead of replying to an earlier mail thread, they might reply to this notification. Then the application isn't able to identify the original SR because it doesn't contain the tags at the end of the notification. So the application considers it as a new SR. The same issue occurs if someone forwards that notification back to the application.

To prevent the application from creating new SRs in such situations, you must do the following. At the end of the message content of your outbound email notification, ensure that you add either the message ID in the format `{##MessageId##}` or the SR number in the format `{##SrNumber##}`.

Note: You can verify the exact format of these tags from any sample acknowledgment email that's sent after an SR is created from an inbound email.

How can I facilitate customers to reply to email notifications created using object workflow?

When you're creating the Email Notification action on an object workflow in Application Composer, ensure that you set the **Reply To Address** to the email address of a channel that's defined in the application. You would specify this email address in the **Specific email addresses** field. For example: `support@mycompanydomain.com`. Customers can then reply to these email notifications created using object workflow.

How can the errors be logged and tracked when a service request is not created through email?

If an email is received by Oracle Engagement Cloud but fails to create a service request due to validation issues on the SR, or for another reason, the failure is captured in the InboundMessage object.

Go to Service Infolets > Inbound Messages to see if a failure occurred. To see the reason for the failure, export the data from the Inbound Messages object, then check the StatusDetails attribute which shows the reason of the failure.

Notes

How do I enable users to edit their internal notes and customer entry messages?

You may want to allow an agent to edit messages they've authored of Type Internal Notes or Customer Entry.

For this option to be available to your users, you must enable the SVC_ENABLE_MESSAGE_CORRECTION profile option. The default value of this profile option is No.

To enable the SVC_ENABLE_MESSAGE_CORRECTION profile option:

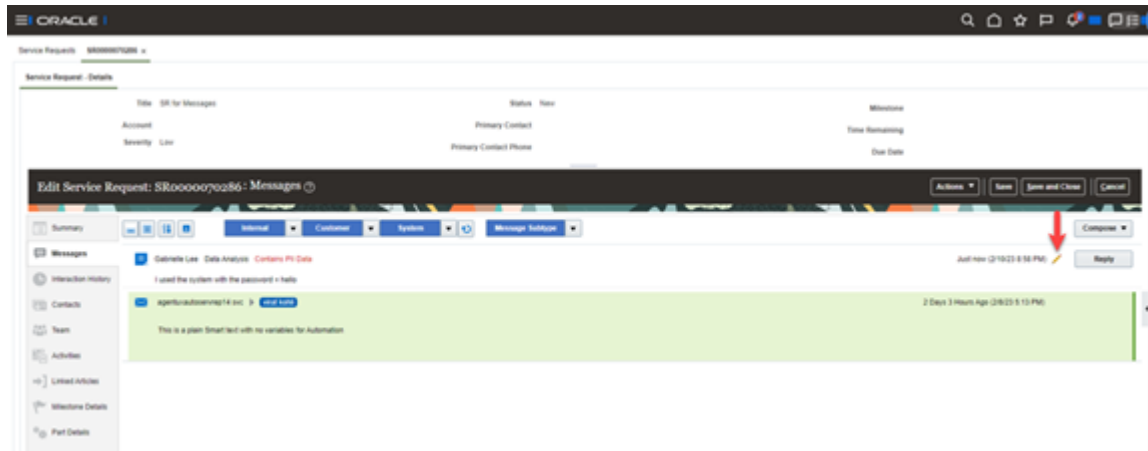
1. Sign in as a setup user or administrator.
2. Navigate to the Setup and Maintenance work area.
3. Open the **Tasks** panel tab and click **Search**.
4. Search for and select the **Manage Administrator Profile Values** task.
5. On the Manage Administrator Profile Values page, navigate to the **Search** region.
6. In the Profile Option Code field, type **SVC_ENABLE_MESSAGE_CORRECTION**, and then click **Search**.

The SVC_ENABLE_MESSAGE_CORRECTION profile option is displayed in the **Search Results: Profile Options** region.

7. In the Profile Option Levels region, set the **Profile Value for the Site Profile Level** as **Yes**.

8. Click **Save**.

After setting this profile option to **Yes**, the user will see a pencil icon for any Internal Notes and Customer Entry Messages they have Authored as shown in the following graphic:



Enable Users to Mark a Message as Containing Personal Information (PII) and Set the Message Subtype Value

You may want to allow an agent to mark a message as containing Personal information and provide additional classification by setting the subtype value on the message.

For these options to be available to users, you must enable the `ORA_SVC_SHOW_ADDITIONAL_FIELDS_IN_MESSAGES` profile option. The default value of this profile option is None.

To enable the `ORA_SVC_SHOW_ADDITIONAL_FIELDS_IN_MESSAGES` profile option:

1. Sign in as a setup user or administrator.
2. Navigate to the Setup and Maintenance work area.
3. Open the **Tasks** panel tab and click **Search**.
4. Search for and select the **Manage Administrator Profile Values** task.
5. On the Manage Administrator Profile Values page, navigate to the **Search** region.
6. In the Profile Option Code field, type `ORA_SVC_SHOW_ADDITIONAL_FIELDS_IN_MESSAGES`, and then click **Search**.

The `ORA_SVC_SHOW_ADDITIONAL_FIELDS_IN_MESSAGES` profile option is displayed in the **Search Results: Profile Options** region.

7. In the Profile Option Levels region, set the **Profile Value for the Site Profile Level** to one of these values:
 - o **Both:** Show both PII and Subtype fields in the message header.
 - o **Message Subtype:** Show just the Subtype field in the message header.
 - o **None:** The default value. Do not show any additional fields in the message header.
 - o **PII:** Show only PII field in the message header.

Manage Administrator Profile Values ⓘ

Search : Profile Option

Profile Option Code: ORA_SVC_SHOW_ADDITIONAL_FIELDS Application: [Dropdown]
 Profile Display Name: [Text] Module: [Dropdown]
 Category: [Dropdown]

Search Results

Search Results : Profile Options

View ▾

Profile Option Code	Profile Display Name	Application	Module	Start Date	End Date	Description
ORA_SVC_SHOW_ADDITIONAL_FIELDS	Show additional fields in Messages	Service	Service	6/18/21		The set of box subtype and PII fields in messages

ORA_SVC_SHOW_ADDITIONAL_FIELDS_IN_MESSAGES: Profile Values

Actions: View ▾ [Icons]

Profile Level	Product Name	User Name	Profile Value
Site			[Dropdown: None, Both, Message Subtype, PII]

8. Click **Save**.

Configure Message Sub Type Values

You can create and select subtype values for the different message types that will be shown in the message subtype field when it is exposed in the message header.

To configure these message subtype values do the following:

1. Sign in as a setup user or administrator.
2. Navigate to the Setup and Maintenance work area.
3. Open the **Tasks** panel tab and click **Search**.
4. Search for and select the **Manage Service Request Message Types and Subtypes** task.

5. Create new records for the message subtype and then map them to the appropriate message type by moving them into the **Selected Subtypes** panel as displayed in the following example:

ORACLE

Manage Service Request Message Types and Subtypes ?

Map Subtypes to Types

Types

Chat Transcript
Customer Entry
Forward
Internal Note
Response
System Note
System Response
Wrap Up

Available Subtypes

Used Elsewhere

Selected Subtypes

Data Analysis
Msg SubType_CustomerEntry
Msg SubType_Response

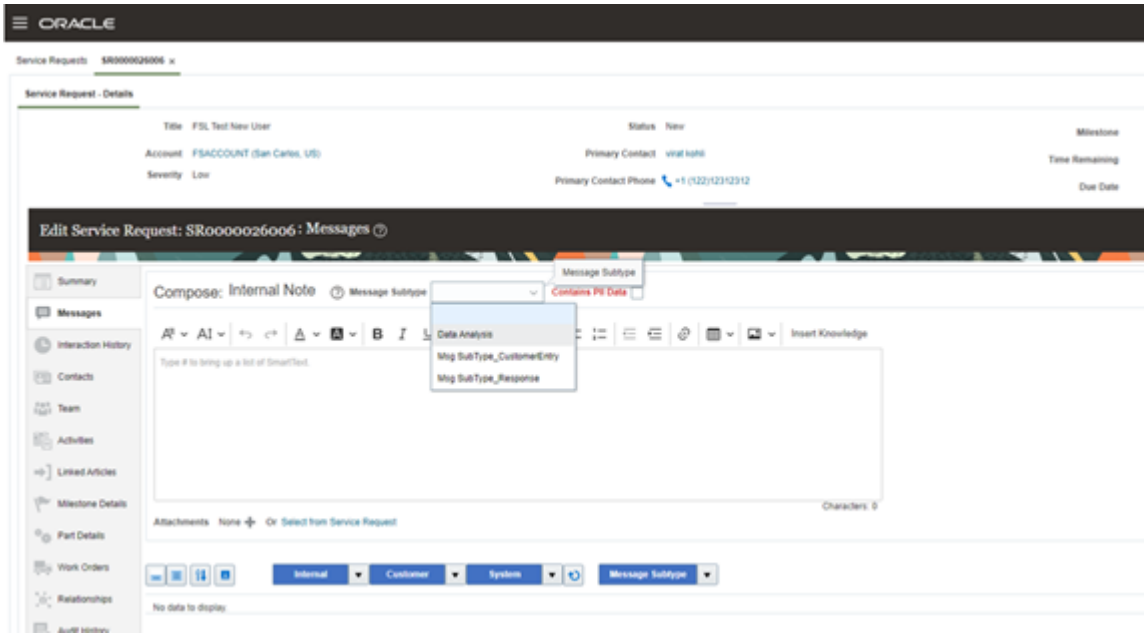
Edit Types and Subtypes

Message Type Code Message Subtype

Actions View Format + X Freeze Detach Wrap

Lookup Code	Display Sequence	Enabled	Start Date	End Date	Meaning	Description
SVC_RESPONSE_TEST	10	☒	m/d/yy	m/d/yy	Msg SubType_Response	Msg SubType_Response
SVC_CUSTOMERENTER_TEST	20	☒	m/d/yy	m/d/yy	Msg SubType_CustomerEntry	Msg SubType_CustomerEntry
DATA_ANALYSIS	30	☒	m/d/yy	m/d/yy	Data Analysis	Data Analysis

And here's what the agent will see when these fields are exposed in the message header:



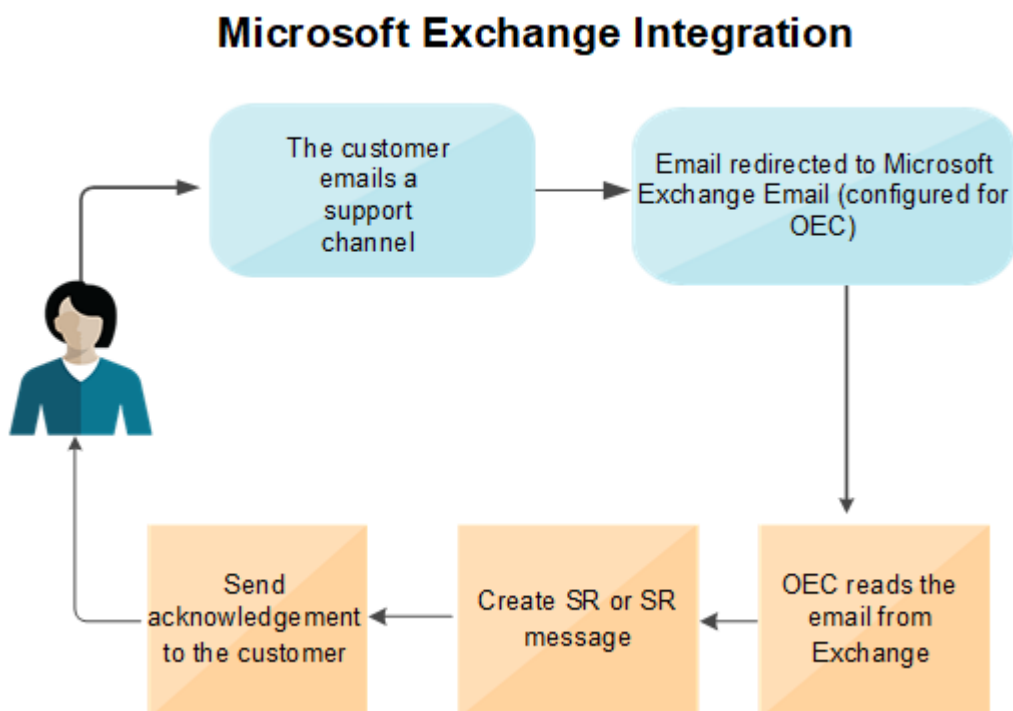
16 Integrate with Microsoft Exchange Server

Overview of the Integration with Microsoft Exchange Server

Integrating with Microsoft Exchange Server enabled you to streamline your service email delivery giving more options and fewer potential failure points.

Create one redirect rule to one of your support mailboxes which is then monitored by Fusion Service and the Microsoft Exchange API.

The following graphic shows a high level overview of the integration:



Note: These procedures require certain set up tasks be done in Microsoft Exchange Server. For information on this, you must consult the Microsoft documentation.

Register an App in Microsoft Azure

You begin the integration by creating and registering an app in Microsoft Azure.

When you set up this app you grant all the required permissions to read emails from support mailboxes. After the registration process, the client ID and secret that you created for the app are securely stored in Oracle Platform Security Services (OPSS).

You must have administrator access to Microsoft Azure and Exchange portals.

Here's how you register the app:

1. Sign in to Microsoft Azure as an administrator.
2. In the explorer list, navigate to **Azure Active Directory > App Registrations**.
3. Click **+ New registration**.
4. In the **Register an application** screen, enter a name for the app, and then select **Single tenant**.

Note: In future releases, multitenant support will be available, but not currently.

The screenshot shows the 'Register an application' page in the Microsoft Azure portal. The breadcrumb navigation is 'Home > zpss | App registrations >'. The title is 'Register an application'. There is a required field for 'Name' with a description: 'The user-facing display name for this application (this can be changed later)'. Below this is a section for 'Supported account types' with the question 'Who can use this application or access this API?'. Three radio button options are present: 'Accounts in this organizational directory only (zpss only - Single tenant)' (selected), 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', and 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)'. A link 'Help me choose...' is provided. Below is the 'Redirect URI (optional)' section with a description: 'We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.' There is a dropdown menu set to 'Web' and a text input field containing 'e.g. https://myapp.com/auth'. At the bottom, there is a checkbox for 'By proceeding, you agree to the Microsoft Platform Policies' and a blue 'Register' button.

5. When you're finished, click **Register**.

6. On the Stage screen, copy and retain the **Application (client) ID**, and **Tenant ID** for later use.

7. In the explorer, click **Certificates & secrets**.
8. Create a new client secret and select **Never** as the expiration option, then copy the client secret for later use.
9. In the explorer, click **API permissions**.
10. Click **Add a permission**, then select **Microsoft Graph**.

11. On the **Request API Permissions** screen, select **Application Permissions**.
12. In the **Select Permissions** field enter **User** and then scroll down and select **User.Read.All**.

13. Grant the following API permissions:
 - Mail.Read
 - Mail.ReadBasic
 - Mail.ReadBasic.All
 - User.Read
 - User.Read.All
 - Mail.ReadWrite
14. Now, click **Grant admin consent required** and select **Yes** for each of the permissions you just added.

The app has now been given permission to read emails.

Limit Access of Mailboxes Using REST API

To provide access to a selected list of outlook mailboxes using the MS Graph API, you now must configure a connection to connect to Microsoft Exchange online PowerShell.

Use the following steps to connect to the PowerShell using docker, then perform the steps to limit the mailbox access.

1. Execute the following commands to install the dependencies to connect to exchange online:
 - `--Install-Module -Name PSWSMan` (If prompted, accept PSGallery as the source for the cmdlets.)
 - `--Install-WSMan` (If prompted, accept PSGallery as the source for the cmdlets).
 - `--Install-Module -Name ExchangeOnlineManagement`
 - `--Import-Module ExchangeOnlineManagement`
 - **Note:** If you don't have access to the PowerShell tool, run the following PowerShell docker image:

```
docker run -it mcr.microsoft.com/powershell
```

2. Execute the following command to connect to Exchange Online: `Connect-ExchangeOnline -Device`

The output will have a URL with an authentication code, which must be opened in a browser.

3. From the browser, authenticate by logging in to the account.
4. Create a new mail-enabled security group consisting of the mailboxes that should be allowed to access: <https://docs.microsoft.com/en-us/exchange/recipients-in-exchange-online/manage-mail-enabled-security-groups>
5. Execute the following command with the required tenantId and the security group email address:

```
New-ApplicationAccessPolicy -AppId <client_Id> -PolicyScopeGroupId <security_group_email_id> -  
AccessRight RestrictAccess -Description "<description>"
```

For example: `New-ApplicationAccessPolicy -AppId d2abe981-a2f3-44f7-af10-acd1c30e61d4 -PolicyScopeGroupId testdistributionlist@4development11395.onmicrosoft.com -AccessRight RestrictAccess -Description "Restrict this app to members of distribution group Test group."`

6. Use following command to verify that the app has permission to access the given mailbox:`Test-ApplicationAccessPolicy -Identity <email_address> -AppId <client_Id>`

For example: `Test-ApplicationAccessPolicy -Identity service_test11@4development11395.onmicrosoft.com-AppId d2abe981-a2f3-44f7-af10-acd1c30e61d4`

The process can take up to an hour to reflect in the REST APIs. Once the change is reflected, an error message similar to the following will be shown, when trying to access a mailbox that is not permitted:

```
"body": {  
  "error": {  
    "code": "ErrorAccessDenied",  
    "message": "Access to OData is disabled."  
  }  
}
```

Now you add these configurations to your Fusion application.

Configure Fusion Email

Now you can configure Fusion email.

1. Sign in to Fusion Service as an administrator.
2. From Setup and Maintenance, click the **Tasks** icon, and click the **Search** link.
3. In the search field, enter **Manage Administrator Profile Values**, and then select it.
4. In the Profile Option Code field, enter `ORA_SVC_ENABLE_INBOUND_EMAIL_EXTERNAL_APPS_CONFIGURATION` and click **Search**.

Manage Administrator Profile Values ?

▲ Search : Profile Option

Profile Option Code

Profile Display Name

Category

Application

Module

5. In the **Profile Values** list, set the profile value to **Yes**.

ORA_SVC_ENABLE_INBOUND_EMAIL_EXTERNAL_APPS_CONFIGURATION: Profile Values

Actions ▾ View ▾ + ×

Profile Level	Product Name	User Name	Profile Value
Site			Yes No Yes

6. Click **Save and Close**.
7. In the Search field, enter **Manage Email**.

ORACLE

Search

Manage Email

Match With [Tasks](#), [Task Lists](#), [Business Objects](#)

Name

- Manage Email Configuration, Registration, and Validation
- Manage Email Profile Options
- Manage Email Profiles
- Manage Email Templates
- Manage Email Templates for HR Help Desk
- Manage Email Templates for Internal Help Desk

8. Choose the task **Manage Email Configuration, Registration, and Validation**.

9. Scroll down to the **Configure External Applications** area and enter the values retained from the setup you did in Azure, including Tenant ID, Client ID, Client Secret and so on.

Configure External Applications

* Tenant ID	d271d316-e042-422f-b36e-037266067193
* Client ID	d2abe981-a2f3-44f7-af10-acd1c30e61d4
* Client Secret	5xm8Q~Bc9CEwNA5UknxAlcUbj2of7PscC_Oy7agq
* Email Address	service_test5@4development11395.onmicrosoft.com
* Activation Date	11.22.2022 AM 01.04.09 

10. Enter the email address you want to use.

Note: You can only set up one email address, such as support.examplecompany.com.

11. In the **Activation Date** field, enter the date you want the processes to start.
12. Click **Save**, and then scroll to the top of the page and click **Done**

Set Inbound Email Attachment Size

The default attachment size for inbound email is 15MB. To increase this size you set configure the SVC_INBOUND_EMAIL_MAX_ATTACH_SIZE profile option. Here's how you do it:

1. Sign in to Fusion Service as an administrator.
2. From Setup and Maintenance, click the **Tasks** icon, and click the **Search** link.
3. In the search field, enter **Manage Administrator Profile Values**, and then select it.
4. In the Profile Option Code field, enter SVC_INBOUND_EMAIL_MAX_ATTACH_SIZE and click **Search**
5. In the Profile Values list, for the Site Profile Level increase the **Profile Value** to the MB value you require, then click **Save and Close**.

How do I create a job to read messages?

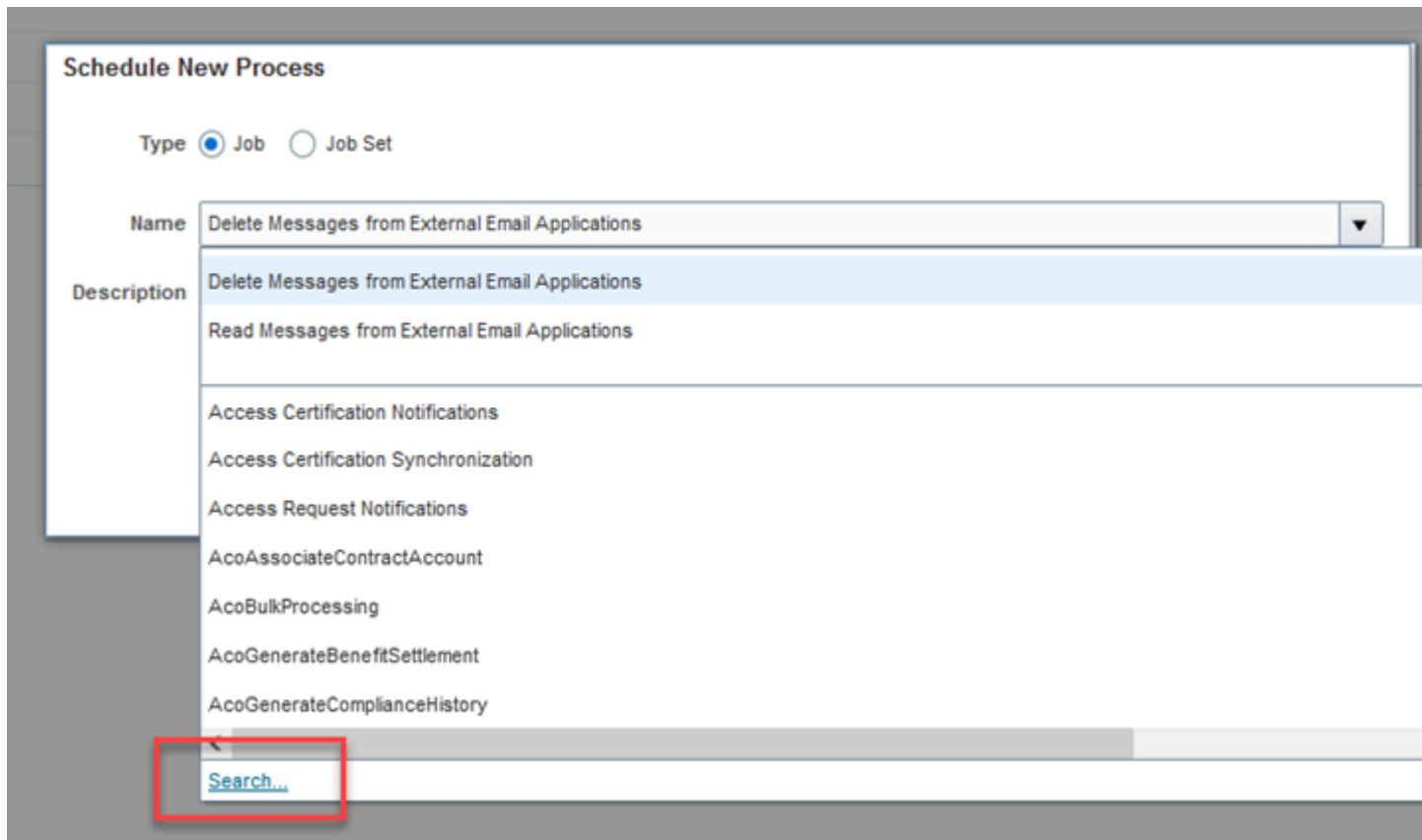
Now, you create an Oracle Enterprise Scheduler (ESS) job to periodically read the email messages from the configured outlook email address.

This process stores all the required details of an email in your database. After reading the required number of emails in a batch, a new event is generated to create an SR or an SR message using the existing SOA composite.

Here's how you create the new job:

1. From the Navigator, select Tools, then **Scheduled Processes**.
2. Click **Schedule New Process**.

3. From the Schedule New Process dialog box, click the drop-down list, and then click the **Search** link.



4. On the Search and Select: Name dialog box, enter any part of **Read Messages from External Email Applications** in the Name field, and then click **Search**.
5. From the results, select **Read Messages from External Email Applications** and click **OK**, and then click **OK** again.
6. On Process Details page, click **Advanced**, then in the **Advanced Options** area, select **Using a schedule**, then set a time, such as every 1 minute, every 2 minutes, or whatever time increment suits your needs.
7. When you're finished, click **Submit**.

Create a Job to Delete Messages

Your mailbox serves as staging area, but messages don't need to be retained there. Use this topic to create a job to periodically delete messages.

1. From the Navigator, select Tools, then **Scheduled Processes**.
2. Click **Schedule New Process**.
3. From the Schedule New Process dialog box, click the drop-down list, and then click the **Search** link.
4. On the Search and Select: Name dialog box, enter any part of **Delete Messages from External Email Applications** in the **Name** field, and then click **Search**.
5. From the results, select **Delete Messages from External Email Applications** and click **OK**, and then click **OK** again.

6. On Process Details page, click **Advanced**, then in the **Advanced Options** area, select **Using a schedule**, then set a time, such as every 1 minute, every 2 minutes, or whatever time increment suits your needs.
7. When you're finished, click **Submit**.

How do I schedule deletion of inbound emails that were previously retrieved from Exchange Server?

You can set up a scheduled process to automatically delete inbound emails from the inbox on your Microsoft Exchange server if those emails have already been retrieved and processed. This job increases performance if you're integrated with Microsoft Exchange Server, by removing unnecessary processing.

Here's how you do it:

1. In Setup and Maintenance, do the following:
 - Offering: Service
 - Functional Area: Communication Channels
 - Task: Manage Email Configuration, Registration and Validation
2. In the Configure External Applications section, select the **Delete Message** check box.

This ESS job, **Delete Messages from External Email Applications**, deletes inbound email from the inbox on Exchange Server, and it will now automatically run according to the schedule set by the administrator.

17 Interactions

Interactions

Overview of Interactions

An interaction is recorded when a communication occurs with customer contacts by means of any communication channel such as email, or phone, or chat.

Web isn't a ready-to-use supported channel for Interactions. Currently, there is limited support for custom objects and channels.

Interactions help you track inbound and outbound communications with your users across any channel. An interaction is typically created for each outbound and inbound communication. An outbound communication is initiated by a company employee such as a salesperson by a phone call, or marketing manager by means of an email. An inbound communication is initiated by a prospect or contact by means of inbound phone call or email.

An interaction tracks whether a communication happened, the channel used for communicating, and any associated business objects that were created or edited during that communication. You can use the Activity Stream to access past wrap up records.

Interactions help track the following ready to use attributes:

- Contact
- Account
- Channel
- Start Time
- End Time
- Resource
- Duration
- Direction
- Associated Business Objects
- Parent Interactions

An interaction can be associated to the following objects:

- Account
- Contact
- Service Request
- Activity
- Work Order
- Lead
- Opportunity

The Interaction History view subtab displays for the following objects by default: Service Request, Account, and Contact objects.

The View Interactions privilege (SVC_VIEW_INTERACTIONS_PRIV) is required for users to view interactions. This privilege is available by default to the Sales and Service Representative, Manager and Administrator roles.

To display the **Service Request** subtab under the Account and Contact details pages, the implementation status of the Service offering must be **Implemented**.

If you need to check your implementation status, here's how you do it:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: **Service**
2. Click **Actions > Edit Implementation Status**.
3. In the **Edit Implementation Status: Service** dialog box, select the **Implemented** option.
4. Click **Done**.

Related Topics

- [Configure Chat Wrap Up](#)
- [Configure Cross-Channel Interactions](#)
- [What's the life cycle of a service request?](#)
- [How you View Interaction History for the Service Request](#)

Configure Cross-Channel Interactions

Overview of Cross-Channel Interactions

Cross-channel interactions enable the administrators to track and manage inbound and outbound interactions that cross from one channel to another.

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.

Note: For cross channel connections to function properly, the account, contact, and resource must be a match with the associated record. For example, if you have a service request where the primary contact isn't the same as the person on the phone, the record won't connect through interactions.

Configure Cross-Channel Interactions

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

The **Manage Setup and Configuration for Cross-Channel Interactions** 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. In the Setup and Maintenance work area, select the **Tasks** icon.
2. Search for the following string:

Manage Setup and Configuration for Cross-Channel Interactions
3. Click the **Manage Setup and Configuration for Cross-Channel Interactions** link.
4. The **Enable Automatic Association of Objects with Interactions** field is selected as **Yes** by default. This enables automatic association of the objects with interactions. When you select **No**, business objects aren't associated with interactions automatically.
5. 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. If the Enable automatic association option is set to No, setting the time here's irrelevant.

6. 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. As with the previous setting, if the Enable automatic association option is set to **No**, setting the time here is irrelevant.

If the Enable automatic association option is set to **No**, setting the time here is irrelevant.

Note: When you set both the periods **Search Retrieval Range Prior to the Creation Time** and **Search Retrieval Range Prior to the Last Update Time**, then a query is run and an association is made for one or both the periods based on the match.

CAUTION: Once you start an interaction, it's bound by the time settings you have configured. In other words, if you've set the interaction time period to be five minutes and the call lasts ten minutes, the last five minutes of the interaction will not be recorded.

7. In the **Automatically Associated Objects** section, you can configure whether to automatically associate service requests with interactions. These are set to **Yes** by default.

Out of the box, you can select the objects shown in the table in the Cross-Channel Options topic which follows this topic.

8. Click **Save** or **Save and Close**.

Cross-Channel Options

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

Cross-channel Option	Description
SVC_INTERACTION_AUTO_ASSOCIATION_ENABLE	Enables automatic association of objects to an interaction. It is set to Off by default.

Cross-channel Option	Description
	Note: During bulk importing of data, enabling this option to Yes may result in associating multiple objects to the open interactions. This results in a large number of imported CRM objects getting associated with a single interaction, and leads to distorted data. To avoid this situation, the option may be set to N.
SVC_INTERACTION_CREATION_DATE_INTERVAL_FOR_AUTO_ASSOCIATION	Sets the period, in minutes, to go back from the current time while finding a potential interaction based on creation date
SVC_INTERACTION_LAST_UPD_DATE_INTERVAL_FOR_AUTO_ASSOCIATION	Sets the period, in minutes, to go back from the current time while finding a potential interaction based on last updated date.
SVC_INTERACTION_ENABLED_OBJECTS - (for service request)	- For Service Requests, set this to the value SVC_SERVICE_REQUESTS. - For Activities, set this to the value: ZMM_ACTIVITIES. - For Opportunities, set this to the value: MOO_OPTY. - For Work Orders, set this to the value: SVC_WORK_ORDERS. - For Leads, set this to the value: MKL_LM_LEADS.

Data Security Policies for Interactions

How do I define data security policies for custom roles?

There are data security policies available for interactions which enable you to do the following:

- Provide self-service users a view of interactions where they're the primary contact.
- Provide service users a view of all service interactions.
- Provide help desk users a view of all help desk interactions.

These policies are available with the predefined duty roles as listed in the following table. You can also add these policies to custom roles.

The following table lists the data security policy Service Interaction Header and its enforcement across predefined duty roles.

Duty Role		Business Object	Condition
Customer Self-Service Usage	Can Manage	Service Interaction Header	For all my interactions
Service Request Administrator	Can Manage	Service Interaction Header	For all CRM interactions
Service Request Channel User	Can Manage	Service Interaction Header	For all CRM interactions

Duty Role		Business Object	Condition
Service Request Contributor	Can Manage	Service Interaction Header	For all CRM interactions
Service Request Power User	Can Manage	Service Interaction Header	For all CRM interactions
Service Request Troubleshooter	Can Manage	Service Interaction Header	For all CRM interactions
Conversation Message Management	Can Manage	Service Interaction Header	For all CRM interactions
Conversation Message Administration	Can Manage	Service Interaction Header	For all CRM interactions
HR Help Desk Administration	Can Manage	Service Interaction Header	For all human resource interactions
HR Help Desk Service Request Management	Can Manage	Service Interaction Header	For all human resource interactions
HR Help Desk Analysis	Can Manage	Service Interaction Header	For all human resource interactions
Internal Service Request Administration	Can Manage	Service Interaction Header	For all human resource interactions
Internal Service Request Analysis	Can Manage	Service Interaction Header	For all human resource interactions
Internal Service Request Management	Can Manage	Service Interaction Header	For all human resource interactions
HR Service Request Analysis	Can Manage	Service Interaction Header	For all Help Desk interactions
HR Service Request Administration	Can Manage	Service Interaction Header	For all Help Desk interactions
HR Service Request Management	Can Manage	Service Interaction Header	For all Help Desk interactions

If you've created a custom job role, you must add one of the Data Security Policies mentioned in the previous table.

Here's how you do it:

1. On the **Roles** tab of the Security Console, search for and select your custom role.
2. In the search results, click the down arrow for the selected role and select **Edit Role**.

The Edit Role: Basic Information page is displayed.

3. Click the **Data Security Policies** train stop.
4. Click **Create Data Security Policy**.

The Create Data Security Policy dialog box is displayed. A Start Date value is automatically assigned to the policy but can be changed.

5. In the **Policy Name** field, enter a policy name.

The names of predefined data security policies begin with the words Grant on.

6. Search for and select the database resource **Service Interaction Header**.
7. In the Data Set field, select **Select by instance set**.
8. Select a Condition Name.

-
9. In the **Actions** field, select the actions to which this data security policy applies.

Note: The Manage, Update and Read Actions are required for Chat or Call Center Agents.

10. Click **OK** to save the data security policy.
11. Click the **Summary** train stop.
12. Click **Save and Close** to save the role.
13. Click **OK** to close the confirmation message.

18 Action Plans

Set Up Action Plans

What are action plans?

Action plans are used to complete a series of steps or a sequence of events to resolve different types of requests.

You can associate action plans to service requests if you need to complete a series of steps or a sequence of events to resolve the request .

You can attach action plans to a many objects, such as service request using an administrator-defined template, or by adding individual actions. Actions can be can be required or optional tasks, basic steps such as tasks, or appointments, or more complex steps such as cases, opportunities, or processes. .

Be careful about spacing. This is the Service Center condition. You must remember that each conditioned sentence or phrase won't show in certain output. Here's what I mean.

You can add Action Plans to the following objects:

- Service Requests
- Cases

Agents can add either solo actions, or action plans. Actions can be required or optional tasks, activities, or appointments.

- **Solo Actions** are individual actions added to an object.
- **Action Plans** are added from an administrator-defined template. Action plans are made up of multiple actions.

By using actions and plans, users have:

- A visual aid to see the progress of a solo action or an action plan
- Any warnings on actions or plans

To use the Action Plans feature, set up actions, templates, and categories in Setup and Maintenance.

In Setup and Maintenance, navigate to Service > Action Plan. If Action Plan isn't displayed, use the **Change Feature Opt In** to activate Action Plans.

The following table shows the name and description of each action plan setup task.

Task	Description	Required
Manage Action Plan Profile Options	Manage profile options for the Action Plans feature.	Yes
Manage Action Plan Actions	Create actions that can be used for solo actions or in action templates. You can specify the category, type, duration, and visibility for each action.	Yes

Task	Description	Required
Manage Action Plan Templates	Create templates to group similar actions to be added to an action plan.	Yes
Manage Action Categories for Action Plans	Action categories are used to facilitate finding actions to add to templates or directly to an action plan.	No
Manage Template Categories for Action Plans	Template categories are used by agents to filter and find templates to apply to an action plan.	No
Manage Mapping of Action Plan Status Values	Manage the global setup for status configuration. You can map task, appointment, or request attributes to an action status at a global level. Action status mapping can also be done at the individual action level.	No
Manage Process Integration for Action Plans	Configure the authentication from Fusion Service to your Process Automation instance. Note: Used for the Run Process through Action Plans feature. A subscription to Oracle Process Automation is required in order to take advantage of this feature.	No
Manage Action Plan Process Metadata	Manage the action plan process metadata. Note: Used for the Run Process through Action Plans feature. A subscription to Oracle Process Automation is required in order to take advantage of this feature.	No

Related Topics

- [How do I manage action plan actions?](#)
- [How do I manage Action Plan templates?](#)
- [Map Action Plan Status Values](#)

Manage Action Plan Profile Options

The following are the available action plan profile options and their descriptions.

Manage the Action Plan profile options in Setup and Maintenance. Navigate to:

- Offering: Service
- Functional Area: Action Plans
- Task: Manage Action Plan Profile Options

The following table shows the action plan profile options and their descriptions.

Profile Option	Description
SVC_ENABLE_ACTION_PLAN	Specify if Action Plan is enabled for service requests .
SVC_ALWAYS_SHOW_ACTIONPLAN_TAB	Enables the default visibility of Action Plan tab in the use SR. If you enable it, the Action Plan tab will always be visible. Otherwise, it will only be visible if the SR has Action Plans. Note: This profile option isn't used for Next Gen.
SVC_ENABLE_SUBSCRIPTION_SCHEDULES_IN_ACTION_PLANS	Enable calculation for action plan estimated completion times based on schedules defined in Subscription Cloud. Note: If you don't set this profile option, contract schedules are used. Oracle recommends using subscription schedules. If an action plan template doesn't have an explicitly assigned valid subscription schedule, then SVC_DEFAULT_CRM_CALENDAR/SVC_DEFAULT_HCM_CALENDAR profile options are used to specify a default contract schedule to use as a calendar. If SVC_DEFAULT_CRM_CALENDAR/SVC_DEFAULT_HCM_CALENDAR aren't set, a 24/7 schedule is used by default to perform any date computations.
ORA_SVC_DEFAULT_SUBSCRIPTION_SCHEDULE	Specifies a default subscription schedule to use as a calendar if subscription schedules are enabled (profile option SVC_ENABLE_SUBSCRIPTION_SCHEDULES_IN_ACTION_PLANS = Y) and the action plan template doesn't have an explicitly assigned subscription schedule. Note: If an action plan template doesn't have an explicitly assigned valid subscription schedule, then a 24/7 schedule is used by default to perform any date computations.
ORA_SVC_AP_PROCESS_METADATA_VALIDATION_ENABLED	Enable the validation of process metadata in action plan. Use this profile option to validate the process being associated to an action when a dynamic process action is created or updated in a REST request.

Profile Option	Description
	Note: Used for the Run Process through Action Plans feature. A subscription to Oracle Process Automation is required in order to take advantage of this feature.
ORA_SVC_AP_DEFAULT_OBJECT_LINK_TYPE	The Object Link Type used when Object Links are created by Actions and Plans.
ORA_SVC_AP_ENABLE_OBJECT_LINK	Enable the creation of object links between supported objects in actions and plans.

How do I manage relationships for action plans?

Action plans can have relationships to other objects. For example, an action can be blocked by another object, or a duplicate of another object.

About Relationships

Agents can create business-driven relationships between service requests and leverage them to make updates across SRs. You must enable this feature in Functional Setup Manager.

Enable Relationships

To enable the Relationships feature:

In the Setup and Maintenance work area go to the following:

- Offering: Service
 - Functional Area: Action Plans
 - Task: Manage Action Plan Profile Options
1. Select the profile value **ORA_SVC_AP_ENABLE_OBJECT_LINK**.
 2. In the drop-down list of the value, select Yes.
 3. Click **Save and Close**.

Now, you can set the type of object link that's used for the relationship. For example, blocked, or duplicate.

1. Click the profile value **ORA_SVC_AP_DEFAULT_OBJECT_LINK_TYPE**.
2. Select the relationship type from the Profile Value drop-down list.
3. Click the **Add (+)** icon to continue adding the relationship types you want to use.
4. When you have finished, click **Save and Close**.
5. Click **Done**.

Manage Link Types

You can manage link types in the Manage Object Link Types task.

In the Setup and Maintenance work area go to the following:

- Offering: Service

- Functional Area: Service Request
- Task: Manage Object Link Types

To edit or disable the predefined object link type:

1. Click the link type in the list.
2. Make your changes. To disable, deselect the disable check box.
3. Click **Save and Close**.

To create a link type:

1. Click **Create Link Type**.
2. Enter all the required information:
 - Name
 - Short Name
 - Outward Value
 - Inward Value
3. Select the check box to enable the type.
4. Click **Save and Close**.

How do I manage action plan actions?

Use the Manage Action Plan Actions task to create and edit actions for action templates or individual actions used in service requests. In this task, you specify the category, type, duration, visibility, and attribute mappings for each action.

To manage action plan actions, in the Setup and Maintenance work area go to the following:

- Offering: Service
- Functional Area: Action Plan
- Task: Manage Action Plan Actions

Create a New Action

Here's how to create a new action:

1. Click **Create Action**.
2. Enter the action name.
3. If you enabled business units for service requests for help desk requests or stripe codes, use the drop-down lists to select the stripe code and BU.
4. Select **Category**.

Select **Class**.
5. Select the **Action Type**.

Depending on the Fusion Service features you're using, Action types can include:

- Appointment

- Article

Note: You can create an Action of type Article. You must select Stripe Code of the action and then select Sub Type of the action. The Attribute Mapping section must be populated as it's required for any other type of actions.

- Case
- Dynamic Process
- HR Help Desk Request (If you're using the Help Desk Offering)
- Internal Service Request (If you're using the Help Desk Offering)
- Opportunity (If you're using the Sales offering)
- Service Request
- Task

Follow the next four steps if you're creating a Dynamic Process Action (with the Run Processes Through Action Plans feature). If you aren't using this feature, you can skip the next four steps.

6. Click the **Edit** icon in the **Process Name** field.

7. Search for and select the process you want to add to the action.

Processes must already be created in Oracle Process Cloud and your connections to Oracle Process Cloud must be working.

Also, the administrator must be added to the Process Application Administrator role in the process workspace UI so they can see all processes.

Note: When you create processes, you must define the following case-sensitive input parameters - objectNumber, objectType, actionPlanId, actionPlanActionId, and processParentObjectType when configuring dynamic processes. For Solo Actions, input parameter `processParentObjectType` will have value 'ActionPlan' while for Plan based actions it will have 'ActionPlanAction'.

8. Click **OK**.

9. Select to use the default version of the process, if you want to use the default process version instead of the version-specific process.

CAUTION: After a process is deployed, overriding it will remove all running instances. Oracle recommends that you create a new version for more changes and mark it as default, if it's required. An action which has **Use default version** selected will ensure that only the default version in Oracle Process Automation is used to create process instances. For processes whose inputs are form based, make sure the interface argument is the same.

10. Select the **Action Visibility**.

Visibility options include:

- Not Published - Only visible to the administrator. The action isn't available to add to a template and can't be added by an agent as an another action.
- Customer Visible - Optional. Used only for user interface configurations to display or hide Actions.
- Internal Only - Visible internally only.

11. Enter a context for the action.

Context options include:

- Opportunity
 - Service Request
 - Internal Help Desk Request
 - HR Help Desk Request
 - Case
12. Enter a numeric value for the **Duration** of how long the task should take. Select a unit of measure for the numeric value (days, hours, or minutes).
 13. Select a **Stripe Code** (CRM, HCM, ISR, or HRSR) from the drop-down list.
Note: If you select the type as SR, you shouldn't select stripe as ISR/HRHD. This leads to ambiguity whether the underlying object created should be CRM SR or ISR/HRHD.
 - If the agent wants the associated object to be CRM SR, select the following:
 - Type: SR
 - Stripe: ORA_SVC_CRM
 - If the agent wants the associated object to be ISR, select the following:
 - Type: ISR
 - Stripe: ORA_SVC_ISR
 - If the agent wants the associated object to be HRHD, select the following:
 - Type: HRHD
 - Stripe: ORA_SVC_HRHD
 14. Select a **Category** from the drop-down list.
 15. You can define a specific relationship to be created for the action by selecting an **Object Link Type** from the drop-down list. If you don't select a specific object link type, the global setting is used.
 16. Check the Copy Attachments check box if you want all attachments to be copied from the context object to the target object. You can select to copy all attachments for the following object types:
 - Cases
 - Service Requests
 - Help Desk Requests
 - Internal Service Requests
 17. Enter a description of the action and any pertinent details.
 18. If necessary, make edits on the **Attribute Mapping** or **Status Mapping** tabs. These are explained next.
 19. Click **Save and Close**.

Attribute Mapping Tab

Every action in an action plan has a related object (as in a task, appointment, opportunity, case, or service request) that's automatically created when the action starts.

The attribute mapping tab is where you set up the mapping of information from the action, parent service request, and user-defined values into the related object when it's created.

Note: There are no attributes to map for Dynamic Processes. If you're creating a Dynamic Process Action, there isn't anything to do on this tab.

Mapped attributes come from the fields in the related object. Required attributes are automatically listed, and you can add optional attributes. The **Mapped To** column shows where the information comes from that populates the field on the related object.

CAUTION: If you map attributes incorrectly, the creation of the business objects during orchestration in Action Plans will fail.

Required fields are shown in the **Required** column. Additional fields might also be listed that can't be changed. For example, Activity Type displays Task if that action type is Task or defaults to Appointment if the action type is Appointment.

Here's how you 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 you selected in the previous step, enter free-form text to the box for free-form values. For service request or action, select the field from the choice list in the last column.

For more information about mappings for an article, see the topic Attribute Mappings for Knowledge Articles in this chapter.

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 object, the status mapping rules decide 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 you don't define the status mapping at the action level, then the default is the global status mapping. Also, there are no statuses to map for Dynamic Processes. If you're creating a Dynamic Process Action, there isn't anything to do on this tab. There are however, predefined status value mappings that can't be changed.

For more information about the global action status configuration, see the topic Manage Mapping of Action Plan Status Values.

The following table shows the predefined Process Automation status values that are mapped with Action Plan status values.

Process State	Action Status
Active	ORA_SVC_IN_PROCESS
Completed	ORA_SVC_COMPLETED

Process State	Action Status
Terminated	ORA_SVC_COMPLETED
Closed	ORA_SVC_COMPLETED

For more information about the global action status configuration, see topic [Manage 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.

Example of a Status Mapping

Let's say you want an action to be marked completed when a service request's status is completed or canceled.

The following table shows the values to enter on the Status Mapping tab.

Column	Value
Status This is the status you want the action to be.	Completed
Attribute	Status

Column	Value
This attribute comes from the SR.	
Operator	is one of
Value This is the request's status. You can select multiple values from the list.	Completed, Canceled

Dependencies Tab

When an action is added to a template, the **Dependencies** tab shows the template name. You use 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 don't affect active action plans that use the template.

Action Errors Tab

The **Action Errors** tab lets you to recreate actions for users when there are errors in action attribute mapping. For example, if the attribute in an attribute mapping is mapped incorrectly, the **Action Errors** tab is displayed. From the tab, you can see all the action plans that the action is used in, and after they're fixed, the action plan actions can be recreated directly from this tab.

This tab isn't used for Dynamic Process actions.

The Action Errors tab shows a table with the action plan number, action plan name, and the status details of the action.

Review the attribute mappings the action and correct the error. After attribute mapping for the action is corrected, you can return to the **Action Errors** tab and click **Recreate**.

Edit an Existing Action

To edit an existing action:

1. In the **Manage Action Plan Actions** task, select the template you want to edit.
2. Edit the action.
3. Click **Save and Close**.

How do I manage 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.

In the Setup and Maintenance work area, go to the following:

- Offering: Service
- Functional Area: Action Plan

- Task: Manage Action Plan Templates

Create a New Action Plan Template

To create a new action plan template:

1. Click **Create Template**.
2. Enter a name for the template.
3. Select the type of business calendar for the duration and end dates of the actions.

Note: The calendar comes from the list of schedules defined on the Subscription Management in the Manage Availability page. For more information about calendars, see the Create a Coverage Schedule topic in this guide.

4. Enter a start date for the template.
5. Enter an end date for the template (optional).
6. Check **Published** to publish the template when it's ready for agents to use.
7. Enter a **Description**.
8. Select a **Category**. Categories are available only if you set them up in the Manage Action Categories for Action Plan task.
9. Select a the **Context** from the list of values. Context values include:
 - Article
 - Case
 - HR Help Desk Request (If you're using the Help Desk offering)
 - Internal Service Request (If you're using the Help Desk offering)
 - Service Request
10. If business units are enabled, select the business unit for the template.
11. If stripe codes are enabled, select the stripe code for the template.
12. Select **Auto Start** to choose the action plan template to start automatically.

Any Templates with this setting, once assigned using the Action Plan REST API, are started either on the next execution of the action plan scheduled job (Monitor Action Plan Actions), or a REST call to refresh the action plan..

Note: You might need to create the action plan with `create_&_initiate` status using the REST API. If you create it using just the Create status, it won't consider it for Auto Start on the REST scheduled job or refresh the REST action call.

13. Add actions to the template by clicking the **Add** icon in the **Action Flow** region of the page.
14. On the **Add Actions to Template** page, search for the action to add by entering a name, category, or action type.
15. Click **Search**.
16. From the search results, add an action by clicking the Add icon for the action you want to add.

17. Now, search and add all the actions you want to place on the template.

Tip:

- You might not see all actions. Actions can be limited by stripe and business unit.
- 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**.

18. When you've 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.

19. 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.
20. 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.

21. 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 can skip the action if it doesn't apply to a specific action plan. Mandatory actions can't be skipped.

22. Click **Save and Close** to complete the template.

Edit 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**.

Related Topics

- [How do I manage action plan actions?](#)
- [How do I create a coverage schedule?](#)
- [How do I update existing setup data?](#)

Map Action Plan Status Values

Use the Manage Mapping of Action Plan Status task 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, in the Setup and Maintenance work area, go to the following:

- Offering: Service
- Functional Area: Action Plan
- Task: Manage Mapping of Action Plan Status Values

On the Action Status Configuration page, select from the following action types.

- Appointment
- Case
- HR Help Desk Request
- Internal Service Request
- Opportunity
- Service Request
- Task

You can edit the existing status, or click the **Add** icon to add additional status configurations.

Note: When assigning user-defined values, you can enter any value as a user defined value during attribute mapping. There's no validation performed on that value.

Related Topics

- [What are action plans?](#)
- [How do I manage action plan actions?](#)
- [Overview of Opportunities](#)

Use Oracle Process Automation with Action Plans

Overview of Using Oracle Process Automation with Actions and Plans

You can use Oracle Process Automation with the Action Plans feature to design processes that:

- Automate your Actions and Plans. You can add automated steps to your Actions and Plans based on conditions you create in Oracle Process Automation.
- Leverage Oracle Process Automation capabilities to:
 - Connect to any system over REST APIs
 - Implement workflow into your actions and plans

The topics in this section take you through the process of enabling and configuring the Run Processes through Action Plans feature.

Note: A subscription to Oracle Process Automation is required in order to take advantage of this feature.

Configure Oracle Identity Cloud Service and Oracle Process Automation

To access the Oracle Process Automation Designer, Workspace, and REST APIs, you must first enable user synchronization in IDCS and assign user roles.

See the steps in the topic Use Oracle Process Automation with Fusion-Based Oracle Cloud Applications in the Using Oracle Process Automation guide.

As an IDCS administrator, make the following role assignments:

- Assign the Service Administrator role to the following groups:
 - Sales Administrator
 - Case Worker
 - Human Resource Help Desk Agent (if using the Help Desk product)
 - Internal Help Desk Manager (if using the Help Desk product)
- Assign the Service Developer role to the following groups:
 - Customer Service Representative
 - Case Worker
 - Human Resource Help Desk Agent (if using the Help Desk product)
 - Internal Help Desk Agent (if using the Help Desk product)

In the Oracle Process Automation Workspace add the following groups to the Oracle Process Application Administrator Global role:

- Sales Administrator
- Case Worker
- Human Resources Help Desk Agent (if using the Help Desk product)
- Internal Help Desk Manager (if using the Help Desk product)

Configure Oracle Process Automation with Oracle Fusion Service

Enable the Run Processes through Action Plans Feature

You can use Oracle Process Automation in Actions and Plans in Fusion Sales and Service version 21.04 or later.

To enable the feature, go to Setup and Maintenance and do the following:

1. Go to the Service offering.
2. Click **Change Feature Opt In**.
3. If it's not already done, select the **Enable** check box for Action Plans.
4. Click the **Edit** icon for Action Plan **Features**.
5. Select the **Enable** check box for Run Processes through Action Plans.
6. Click **Done**.
7. Click **Done** on the Opt In page.

Ensure all users of your process-based Actions and Plans have:

- The SVC_ACTION_PLAN_PROCESS_PRIV privilege provisioned in Sales and Service.
- The required permissions to view and edit their processes in Process Automation.

Manage Oracle Process Automation Integration

To configure authentication from Oracle Service Center to the Oracle Process Automation instance:

1. In Setup and Maintenance, go to the following:
 - Offering: Service
 - Functional Area: Action Plans
 - Task: Manage Process Integration for Action Plans

The Configure Process Integration for Action Plan page displays. This is where you add the proxy user who initiates processes for jobs.

2. Enter the credentials for a user who has the Process Application Administrator role in Oracle Process Automation.
3. Click **Save**.

If successful, the following confirmation messages display.

- Process credentials saved.
- Process credential validation is successful.

4. Once credentials save and successfully validate, click **Save and Close**.

Manage Action Plan Process Metadata

With this task, you can see what current processes are available to be mapped to actions.

In Setup and Maintenance, go to the following:

- Offering: Service
- Functional Area: Action Plans
- Task: Manage Action Plan Process Metadata

In the Process Metadata page, enter search criteria and click the **Search** icon.

There's no action needed for this page. This is a read-only list of processes where you can see all processes by name, their versions and which versions are the default.

Export and Import Action and Plan Configurations

If you need to, you can export and import action plan configurations from you test environment to your production environment.

For export and import of Actions and Plans, refer to *Import Your Action Plan Data* in the *Oracle Fusion Understanding Import and Export Management for Sales and Service* guide.

For export and import of Process Automation processes, refer to the *Process Automation documentation*.

Related Topics

- [Import Your Action Plan Action Data](#)

19 Integrate with Intelligent Advisor

Oracle Intelligent Advisor

Oracle Intelligent Advisor helps service agents deliver enriched, personalized, and precise customer service.

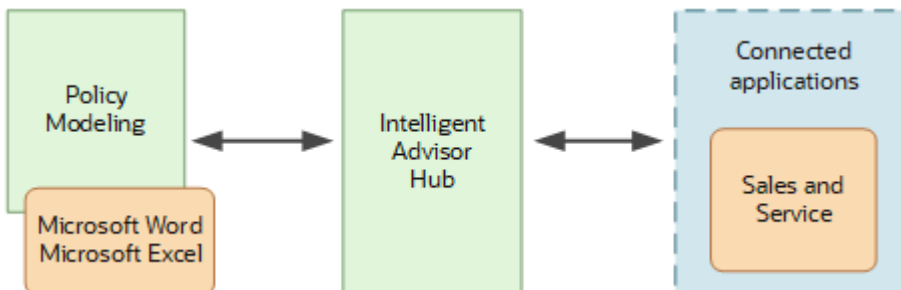
Intelligent Advisor can be connected to Fusion Sales and Service Center for use by agents. Intelligent Advisor interviews can be used for:

- Service requests across a range of industries
- Knowledge streaming by helping agents and contacts identify the most appropriate knowledge article or advice for their needs
- Any multi-page information gathering in Oracle Visual Builder
- Personalized self-service advice with Sales and Service

Using Intelligent Advisor within the Fusion Sales and Service Center interface:

- Agents must be logged in to use Intelligent Advisor interviews
- Specific Sales and Service Center objects can be created and updated

There's desktop modeling software for building interviews called Oracle Policy Modeling. The Intelligent Advisor Hub connects Policy Modeling to Fusion Sales and Service Center.



The steps to integrate Fusion Sales and Service Center with Intelligent Advisor are:

1. Create an Intelligent Advisor Hub.
2. Create a Fusion Sales and Service Center connection in Intelligent Advisor Hub.
3. Download and install Policy Modeling.
4. Create a project in Policy Modeling.
5. Connect the Policy Modeling project to Intelligent Advisor Hub.
6. Map data to and from Fusion Sales and Service Center in Policy Modeling.
7. Deploy and activate the Intelligent Advisor interview.
8. Embed the interview in a Fusion Sales and Service Center agent interface.

Related Topics

- [Intelligent Advisor Documentation Library](#)

Create a Fusion Sales and Fusion Service Connection in Intelligent Advisor

The next step is to create the Fusion Sales and Service Center web service connection in the Intelligent Advisor Hub that your team can use for new or existing interviews. This connection enables the two applications to work together and share information.

The installation administrator should have set up access for the Sales Administrator, or a similar role that administers Intelligent Advisor for your organization. Once you sign in, you create a connection to your Fusion Sales and Fusion Service web services to get access to data for creating or editing interviews.

Related Topics

- [Create a connection to Oracle Sales and Fusion Service](#)

Download and Install Policy Modeling

Once the connection to the web service is established, the next step is to download and install Policy Modeling.

Policy Modeling is the desktop tool modelers use to create and edit interviews, and to map data from Fusion Sales and Service Center to Intelligent Advisor.

Policy Modeling can be downloaded from Intelligent Advisor Hub.

Related Topics

- [Install Policy Modeling](#)

Connect the Policy Modeling Project to Intelligent Advisor Hub

The next step is to connect the Policy Modeling project to Intelligent Advisor Hub which provides access to the Fusion Sales and Fusion Service data model.

In other words, Intelligent Advisor receives data model information from the Intelligent Advisor Hub which in turn receives the data model information from Fusion Sales and Fusion Service. This enables Intelligent Advisor to load and save data to Fusion Sales and Fusion Service.

To connect your Policy Modeling project to Fusion Sales and Fusion Service via Intelligent Advisor Hub:

1. In Intelligent Advisor, connect the project to Intelligent Advisor Hub.
2. Choose the Fusion Sales and Fusion Service connection you created previously.

Related Topics

- [Specify the Intelligent Advisor Hub for a project](#)
- [Choose a connector framework web service connection for a project](#)

Map Data to and from Fusion Sales and Fusion Service to Policy Modeling

Modelers can then map the data in the policy model Intelligent Advisor project to the appropriate tables objects and fields in Fusion Sales and Fusion Service.

Data mapping lets you load data into an Intelligent Advisor interview from Oracle Fusion Sales and Fusion Service, and to save outcomes, including creating new records, from the interview back to Fusion Sales and Fusion Service. The input mappings are used to load values, and the output mappings are used for updating and saving values.

Related Topics

- [Deploy a Policy Modeling project](#)
- [Activate a project](#)

Configure Intelligent Advisor Interviews

How do I create service requests by interview?

You can have your users create service requests through an interview-style format using Oracle Intelligent Advisor. Using an interview to create service requests, can ensure service requests are created only after all critical questions are answered.

To create service requests by interview, you use Intelligent Advisor to create the interview, then Visual Builder Studio to make the interview available to your users.

Note: This feature requires a license for Oracle Intelligent Advisor.

The steps to enable and use the **Create Service Request by Interview** button on the Service Requests List page.

1. Open the Service Request List page in Visual Builder Studio (**service > ec**) and find the Constants tab in the Right Hand Panel.
2. Click the **Constants** tab.
3. Set the values for the these variables on the **Constants** tab:
 - a. showCreateSRByInterviewButton: true
 - b. opaHubURL: Enter the Intelligent Advisor Hub URL where your interview is deployed.
 - c. createSRInterviewDeploymentName: Enter your interview deployment name. (The name of the interview that you want to use to create the service request).
4. Publish the changes.

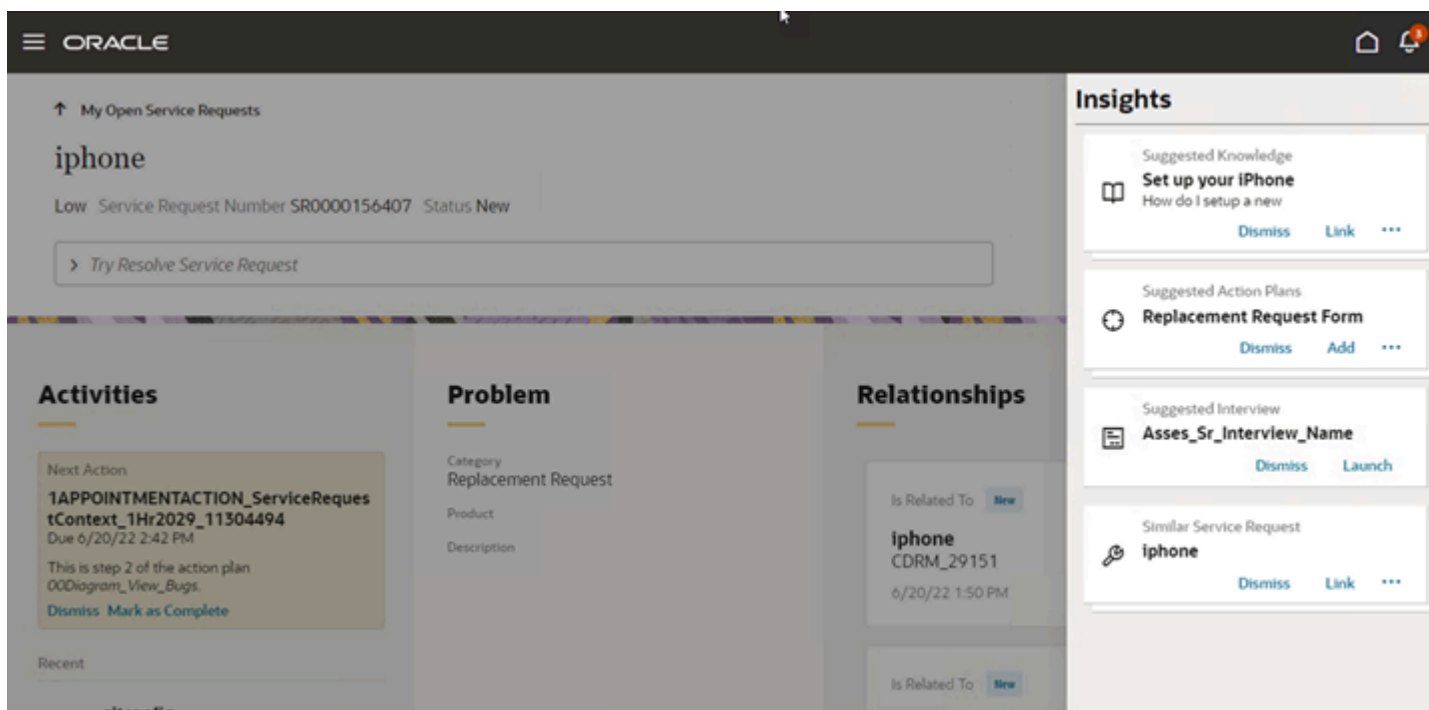
20 Set Up Agent Insights

Overview of the Insights Panel

The Insights panel enables agents to have access to the most commonly used tools, at any time, within the context of each service request they're working on.

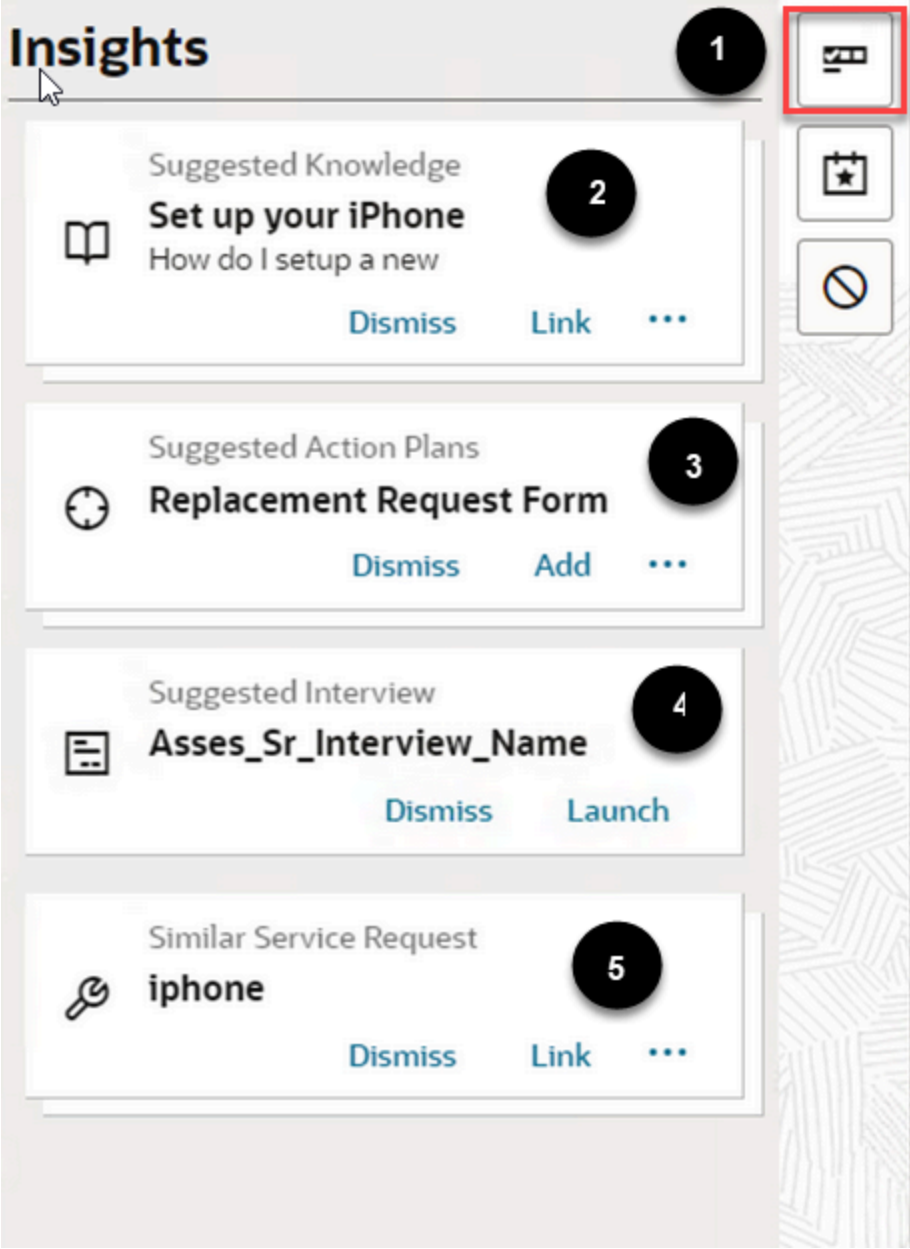
Here's some benefits of setting up this feature:

- Increased consistency in response
- Increased Efficiency
- Reduced Redundancy
- Best practices and standardization can be enforced by deploying companies.
- Agents can access the next-best action quickly to help resolve an SR.



Here's how it works:

An agent working within a service request can view the Insights panel while viewing SR details. The Insights panel displays entries which you configure prior to use. After the Insights panel is enabled, once an agent has selected an SR, the **Insights** button (callout 1) is enabled allowing the agent to click and view the Insights panel at any point in the SR process.



The remainder of the configurable cards are described in the following table:

Callout	Card Type	Actions	VB Variable
2	Suggested Knowledge	• Link. Links the article to the current SR. • Share. Links and opens the Compose Email with the article link added. • Title Click. Open the article in a new browser tab.	knowledgeInsightsCardEnabled Note: This card is displayed only if the View Knowledge Within Service Request role privilege is enabled for the user.

Callout	Card Type	Actions	VB Variable
3	Suggested Action Plans	Add. Adds the suggested action plan to the current SR.	actionPlanInsightsCardEnabled Note: This card is displayed only if the Create Action role privilege is enabled for the user. If the SR doesn't have a category, no action plans are displayed.
4	Suggested Interview	Launch. Opens the Interview page of the current SR, where the SR can be edited using the configured interview.	launchInterviewInsightsCardEnabled Note: This card is displayed only if the Interview is configured.
5	Similar Service Request	Link. Links the suggested SR to the current SR Title Click. Opens the suggested SR in a new browser tab.	similarSrInsightsCardEnabled Note: This card is displayed only if the SVC_ENABLE_SIMILAR_SR profile option is enabled for the site.

Set Up the Agent Insights Panel

You set up the Insights Panel in VB Studio.

Here's how you enable it:

1. In the Service Center application, open an SR to view the Service Request detail layout.
2. Click the **Settings and Actions** menu, and select **Edit Page in Visual Builder**.
3. Choose your project.
4. Expand the Oracle Fusion Service UI Extension App list, and select the **Service** node.
5. In the **Constants** list, select **agentInsightsEnabled**, then in the properties window, change the default value from False to **True**.
6. Now, to configure which cards are displayed within the in the service folder, expand the ec folder, and then expand the sr node.
7. Make sure the following Constants are set to true.
 - o actionPlanInsightsCardEnabled
 - o knowledgeInsightCardEnabled
 - o launchInterviewInsightsCardEnabled
 - o similarSrInsightsCardEnabled
8. Now preview your changes and note the **Insights** button displayed below the Settings and Actions menu.

21 Enable Productivity Tools

How do I enable productivity tools?

Productivity tools such as Smart Text are features you can add to service request create or edit pages.

The following table lists productivity tools that are available for service requests.

Productivity tool	Description	Profile Option Code
SmartText	A reusable fragment of text that you can insert in messages and fields	SVC_ENABLE_STD_TEXT_IN_SR
Keyboard Shortcuts	A combination of keyboard keys that can be used to quickly access actions and buttons on the service request pages.	SVC_ENABLE_KEYBOARD_SHORTCUTS_IN_SR
SR Audit	A subtab that displays the history of service requests from the time they're created. If you use custom roles, you must add the following two privileges to the roles in Security Console for those users to see the SR Audit subtab: - FND_VIEW_AUDIT_HISTORY_PRIV - SVC_VIEW_SR_AUDIT_HISTORY_PRIV	SVC_ENABLE_AUDIT_IN_SR
Severity color coding	A way to color code severity levels so the user can quickly see the SR's severity.	SVC_ENABLE_SEVERITY_COLOR_IN_SR
Knowledge	A list of knowledge articles to help agents find and use the appropriate details.	SVC_ENABLE_KNOWLEDGE_IN_SR
Click to Edit	A way to perform inline editing of records, update records without drilling down, maintain an overview of the SR list pages, and simplify SR data entry.	SVC_ENABLE_CLICK_TO_EDIT

Productivity tools are disabled by default. Here's how you enable them:

1. In the Setup and Maintenance Work area, go to the following:
 - o Offering: Service
 - o Functional Area: Productivity Tools
 - o Task: Manage Global Search Profile Options
2. Search for the profile option code for the productivity tools that you want to add.
3. In the **Profile Values** section for the profile option code, select **Yes** in the **Profile Value** drop-down list.

4. Save the configuration.

Now, you can set up and configure each tool according to your requirements.

Update the Status List Order

The status list on the Summary Details page shows a list of statuses that can be associated with a service request. The order of statuses displayed in the list depends on their assigned ranking. Here's how you update the status ranking:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Service Request
 - o Task: Manage Service Request Status Values
2. Update the **Ranking** column to position the status in the status list according to your requirements.
3. Click **Save and Close**.

How do I manage SmartText entries?

A SmartText entry is a reusable fragment of text that you or your users can create and use in service request messages. You can also create SmartText messages to use in chat.

As an administrator, you can create public SmartText entries that all users can use. Users can't edit these entries.

You can create SmartTexts in multiple languages. You can use rich-text to create SmartText entries in service request message fields by enabling HTML. Other service request fields, such as Problem Description, Solution Description and Transfer Service Request only support plain text. You can also create SmartText entries for chat.

You can create SmartText entries for a specific business unit if you've the HZ_ENABLE_MULTIPLE_BU_CRM profile option enabled (set to Y). With this profile option enabled, the Business Unit drop-down list is displayed in the Manage SmartText UI so you can assign a BU at the time you create a new SmartText. For more information, see Manage Common CRM BU Profile Options for Service BUs.

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.

Required Privileges

The following table shows the different privileges you grant to allow your users SmartText access.

Privilege	Code	Description
Use SmartText	SVC_USE_STD_TXT_PRIV	Allows a user to view, search and insert SmartText.
Manage SmartText	SVC_MANAGE_STD_TXT_PRIV	Allows a user to create, update, and delete SmartText for their own use.

Privilege	Code	Description
Publish SmartText	SVC_PUBLISH_STD_TXT_PRIV	Allows a user to set the scope of SmartText to GLOBAL which allows any user with the correct privilege to view and use that entry.
Administer SmartText	SVC_ADMIN_STD_TXT_PRIV	Allows the user to administer SmartText.

Require Agents to Provide User Entered Values when Inserting SmartText

You can enable your application to require agents to enter values for user variable prompts when they select a SmartText. This ensures that all requested data is captured.

Here's the profile option you need to set to activate the functionality:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Productivity Tools
 - o Task: Manage Global Search Profile Options
2. Search for the profile option code `ORA_SVC_SMARTTEXT_MANDATORY_USER_VARIABLES`
3. Set the site level profile value to **Yes**.
4. Click **Save and Close**.

How do I create and edit private SmartText entries?

Create your own SmartText entries. They're only visible to you, and they're saved in the Private Set tab.

1. On the Create or Edit Service Request page, click the Navigator, choose Tools, and then SmartText.
2. Click the **Create** icon on the **Private Set** tab.
3. On the **Manage SmartText** page, click the **Private** tab.
4. Make sure the default value of **Service** appears in the **Domain** drop-down list.

Note: To create SmartText entries for Chat, make sure to choose Chat from the Domain drop-down list.

5. Make a selection from the **Business Unit** drop-down list (if available).

If the `HZ_ENABLE_MULTIPLE_BU_CRM` profile option is enabled (set to Y) you'll have access to this drop-down list.

If you're an administrator, the drop-down list displays all BUs in your organization. If you're not you're shown only the BUs that you're assigned to along with the Null "No associated business unit" value which is visible to administrators and agents across your organization.

Note: A default business unit is assigned whether you have multiple BUs enabled or not. The default BU is set up when you configure the `HZ_DEFAULT_BU_CRM` profile option.

6. 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.

Note: **Root** is a reserved folder name and administrators can't create a folder with this name.

7. Click the **Create (+)** icon and select **New SmartText**.
8. On the **Manage SmartText** page, enter the following details:
 - Enter a name for the SmartText entry. There are some restrictions on special characters, so see *What are the SmartText naming conventions?*
 - Select **Use HTML** to use rich text formatting in your SmartText entry. You can insert tables and images here.
 - Enter the text you want to insert.
 - Insert a variable, if you want, by typing the `&` symbol followed by the variable name. Click 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.
 - Click **Save as Personal**.
9. To edit the Private SmartText entry, do the following:
 - Select the SmartText entry you want to edit.
 - Select the **Language** of the SmartText entry from the drop-down list
 - You can also create in a new language by clicking the **Add** icon.
 - Edit the Private SmartText entry and click **Save as Personal** to publish the SmartText entry.

You can view and manage all your Private SmartText entries from the Private SmartText tab. You can add, duplicate, and delete your private set SmartText entries.

How do I create and edit public SmartText entries?

Create your own Public SmartText entries. Public entries are available to all users.

To create a Public SmartText entry, do the following:

1. On the Create or Edit Service Request page, click the Navigator, choose Tools, and then SmartText.
2. Click the **Create** icon on the Public tab.
3. On the **Manage SmartText** page, click the **Public** tab.
4. Make sure the default value of **Service** appears in the **Domain** drop-down list.

Note: To create SmartText entries for Chat, make sure to choose Chat from the Domain drop-down list.

5. Make a selection from the **Business Unit** drop-down list (if available).
If the HZ_ENABLE_MULTIPLE_BU_CRM profile option is enabled (set to Y) you'll have access to this drop-down list.

If you're an administrator, the drop-down list displays all BUs in your organization. If you're not you're shown only the BUs that you're assigned to along with the Null "No associated business unit" value which is visible to administrators and agents across your organization.

Note: A default business unit is assigned whether you have multiple BUs enabled or not. The default BU is set up when you configure the HZ_DEFAULT_BU_CRM profile option

6. 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.

Note: **Root** is a reserved folder name and administrators can't create a folder with this name.
7. Click the **Create (+)** icon and select **New SmartText**.
8. On the **New SmartText** page, do the following:
 - o Enter the name for the SmartText entry. There are some restrictions on special characters, so see *What are the SmartText naming conventions?*
 - o Select **Use HTML** to use rich text formatting in your SmartText entry. You can insert tables and images here.
 - o Enter the text you want to insert.
 - o Insert a variable, if you want, by typing the @ symbol followed by the variable name. Click 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.
9. 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.
10. Verify the location where you're saving the SmartText.
11. Select the availability of the SmartText entry from the list of **Availability** options.
12. Click **Publish** to publish the SmartText entry.

To edit a Public SmartText entry, do the following:

1. Select the SmartText entry you want to edit.
2. Select the **Language** of the SmartText entry from the drop-down list
3. You can also create a new language by clicking the **Add** icon.
4. Edit the Private SmartText entry and click **Publish** to publish the SmartText entry.

What are the SmartText naming conventions?

The following restrictions on special characters are applicable when you create a Public or Private SmartText:

- You can't include the following Trigger keys such as # and @ in the name, as these might cause interaction problems.
- The following special characters aren't allowed:
 - o Tilde
 - o Number sign
 - o Percent
 - o Ampersand
 - o Asterisk
 - o Flower Bracket
 - o Backslash
 - o Colon
 - o Angle Bracket

-
- Slash
 - Plus sign
 - Question mark
 - Pipe
 - Quotation mark
 - Double Square Bracket
 - You can't use the following special characters:
 - Exclamation mark
 - Pound
 - Dollar
 - Round Bracket
 - Euro
 - Caret
 - Underscore
 - Equal to
 - Semi-colon
 - Comma
 - Dot

22 Channels for Collaboration

How do I control the display of channels used only for internal conversations?

You can disable the channels that you don't want to be displayed in the application when an agent composes a response message or captures a customer message.

1. Sign in to Service Center as a setup user or administrator.
2. In the Setup and Maintenance work area, go to the **Tasks** panel tab.
3. Search for and click the **Manage Service Request Channel Type Visibility** task.
4. On the Manage Service Request Channel Type Visibility page, if you don't see the channels you want to disable, click Sync on the toolbar of the table.
5. For any channel that you select, you see two rows with different values for **Property Name**:
 - **Enabled for Collaboration**: To enable the selected channel to collaborate with internal users to resolve customer issues, set the **Property Value** for this property name to **Yes**.
 - **Enabled for Customer Interaction**: To enable using the selected channel for interaction of your service agents with your customers, set the **Property Value** for this property name to **Yes**.

Note: You can also use the drop-down list in the toolbar for the **Property Name** field to filter the list by each property name.

6. Set the **Property Value** to **No** to disable the selected channel.

Note:

- For messages of type **Customer Entry** and **Response**:
 - SMS and Slack channels are disabled by default because they aren't supported for customer communications. You can enable or disable the other channels.
- For internal conversations, only the Email, Microsoft Teams, Slack, Web, and custom channel types that you created are supported. You have the option to disable the channel types that you want to hide when starting an internal conversation. For example, if you don't use Slack in your organization, you can disable Slack for collaboration.

7. Click **Save and Close**.

The disabled channels aren't displayed when an agent composes a message.

How do I create an email channel for collaboration or internal conversations?

Agents can have internal conversations on requests when you've set up a separate email channel for internal messages. When the agent wants to start an internal message, they select the internal channel.

To create a separate email channel, create a new channel and set the **Purpose** field for the channel to **Collaboration**.

When starting the internal conversation, the agent must select the internal channel. For example `internal@mycompanydomain.com` Replies appear in the conversation.

For external communications, agents still use the channel you created for the requests. For example `support@mycompanydomain.com` Replies are captured as messages on the request.

Translate the Communication Channel Display Names

Translate Display Name of Communication Channels into Installed Languages

By default, the display name for each of your communication channels is the same in the base language and in all the installed languages. However, the names are translatable, so you can provide language-specific display names in all the installed languages in the application.

To translate the display name of a communication channel into all the installed languages:

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Functional Area: Communication Channels
 - o Task: Manage Communication Channels
2. On the Service Channels page, in the **Account Name** column, click the link to the channel whose display name you want to translate.

The **Edit Channel** window is displayed.
3. Click the Translation Editor.

The **Edit Translations** window is displayed, showing different rows for the channel display name in the base language and the installed languages.
4. Click the row for the first installed language.
5. In the **Display Name** field, specify the selected channel display name in the corresponding installed language.
6. Click **OK**.
7. Starting from step 2, repeat all the steps for each channel.

23 Slack Channels

Overview

Use the Slack Channel for Collaboration

Slack is a team communication tool that facilitates synchronous communication across different devices.

People across departments can use Slack to work together to make better and faster business decisions.

You can set up the integration of Slack with Oracle Service Center, and this helps your company in the following ways:

- Service agents can collaborate with subject matter experts to resolve customer questions and issues faster.
- Notifications are triggered to Slack users and channels when certain events occur on an object. For example, when a service request is set to critical or when an opportunity is closed.

Oracle supports both public facing and workspace private uses of Slack.

For Slack security layers, Oracle supports `https` respecting Oracle Fusion and Slack signing key secret, including Oracle Fusion data security using `pubs` framework (a proxy user framework). The credentials are stored in a credential store. No one has access except the application, but an administrator can configure it.

Slash commands can be configured. There's a configuration page where you can configure slash commands.

APIs are managed through an admin page that interacts with WLD credentials store.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)

How do I set up a Slack channel?

You can integrate your application with Slack in two ways. Depending on your company's requirement, you must follow the steps listed in one of the following sections:

- Set Up a Slack Channel: Method 1: If you plan to use only the predefined slash commands and you don't plan to change existing commands or create your own commands in Slack.
- Set Up a Slack Channel: Method 2: If you plan to create your own slash commands in Slack.
 - See: [How do I set up a Slack channel: Method 2?](#)

Set Up a Slack Channel: Method 1

How You Set Up a Slack Channel for your Fusion Application: Method 1

How you integrate your sales and service application with Slack depends upon your company's requirement. Let's say you plan to use only the predefined slash commands and you don't need to modify the existing commands or create commands in Slack. Then you must complete the steps listed in the following table.

Note: Each of these steps is described in a separate topic, and you must complete them in the given order.

Step no.	Name of topic
1	Install the Fusion Sales and Service App from the Slack Marketplace
2	Update Your Slack App Credentials in Functional Setup Manager
3	Create a Slack Channel in Your Fusion Application
4	Enable Notifications to Slack
5	Add a Slack Contact Point for Your Sales Representatives Who Use Slack (Required for your Sales application, Not Required for your Service application)
6	Set Up a Scheduled Process to Synchronize Slack Users (Required for your Service application, Not Required for your Sales application)

Install the Fusion Sales and Service App from the Slack Marketplace

You can install a preconfigured Fusion Sales and Service Slack app to your workspace.

1. Navigate to: slack.com/apps
2. Ensure that you're signed in to your Slack workspace.
3. On the Slack App Directory page, navigate to the Essential Apps region.
4. Search for and click **Oracle Fusion Sales and Service**.
5. On the Fusion Sales and Service page in the Slack App Directory, click **Add to Slack**.
6. On the page that requests permission to access your Slack workspace, click **Allow**.
7. On the next page, do the following:

- a. Provide a name for the app.
- b. Provide a description for the app.
- c. Change the application icon if required.
- d. Set the Fusion Sales and Service web address to: `https://pod_name:443`

Here, pod_name is the name of the Fusion Sales and Fusion Service instance provisioned for you.

8. Copy the contents of the text box located in the **Add your Slack credentials to your app** region.

Note: These details are used in updating Functional Setup Manager (FSM) with credentials of your Slack app.

Update Your Slack App Credentials in Functional Setup Manager

To update the credentials of your Slack app in the Functional Setup Manager (FSM) in Fusion Sales and Service:

1. Sign in to your Fusion application 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. Search for and select **Manage Collaboration Network Credentials**.
5. On the Manage Collaboration Network Credentials page, specify the **Client Identifier**, **Client Secret**, and **Signing Secret**.

Note: This is the same text that you copied in the last step of the "Install the Fusion Sales and Service App from the Slack Marketplace" topic.

6. Click **Save and Close**.

Create a Slack Channel in Your Fusion Application

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

For each of your Slack workspaces, you must create a Slack channel in Fusion Sales and Service. So if you have multiple Slack workspaces, you have multiple Slack channels in Fusion Sales and Service.

To create a Slack channel in the Fusion application:

1. Sign in to the Fusion application 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. Search for and select **Manage Communication Channels**.
5. In the Service Channels page, click **Create Channel**.
6. In the Create Channel window:
 - The **Stripe Code** is set as **CRM** by default.
 - Select the **Purpose** as **Collaboration**.
 - Select the **Channel Type** as **Slack**.
 - Click **Authorize**.

A new browser tab opens and the following page is displayed in Slack: Sign in to your workspace.

7. In the space provided below the **Enter your workspace's Slack URL** prompt, enter the workspace for which you want to create the Slack channel.
8. Click **Continue**.
9. Sign in to your Slack workspace as an administrator.
10. Complete the next steps on Slack to provide permissions.
11. Click **Authorize**.

A message is displayed, stating that the authorization is complete.

12. Close this browser tab and return to the channel creation page in Fusion Sales and Fusion Service.

Notice that the **Account Name**, **Channel Code**, and **Display Name** are prepopulated.

13. Click **Save and Close** to save the Slack channel.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Enable Notifications to Slack

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

All outbound messages from Fusion Sales and Fusion Service to Slack use the notifications framework. So, for outbound messages to be delivered to Slack, you must enable notifications.

For more information about how to enable notifications, see "Enable Notifications for Service Requests".

Related Topics

- [How do I enable notifications?](#)
- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Add a Slack Contact Point for Your Sales Representatives Who Use Slack

This is a required step for Sales from "How You Set Up a Slack Channel for Your Fusion Application".

(Required for your sales application, Not Required for your service application)

Every Slack user has a unique Slack user ID. When a sales representative executes a slash command in Slack, his information needs to be retrieved from Fusion Sales and displayed in Slack. For this to happen, you must add a Slack contact point in Fusion Sales for every sales representative who uses Slack. This way, you map the unique Slack user ID of each sales representative to his resource record in Fusion Sales.

1. In the Navigator for Fusion Sales, click **Resource Directory**.
2. Click the Tasks panel tab.

3. In the Resources region, click **View Resources**.
4. On the View Resources page, navigate to the **Resource Name** field.
5. Type the name of the resource to whom you want to give the capability to execute Slash commands.
6. Click **Search**.
7. In the Search Results area, click the record of the sales representative.
8. On the Resource page of the sales representative, click the Additional Contact Information tab.
9. Click the plus icon to create a new contact point.
10. In the **Create Contact Point** dialog box, in the **Contact Point Type** drop-down list, select **Social network**.
11. In the **Purpose** field, select **Work**.
12. In the **Social Network** drop-down list, select **Slack**.
13. In the **User ID** field, find and enter the Slack user ID of the service representative as follows:
 - a. In the Slack workspace, in the Direct Messages area, select the name of the sales representative.
 - b. In the adjacent pane, click the name of the sales representative.
 - c. Click **View profile**.
 - d. In the **Workspace Directory** panel that's displayed, click the ellipsis symbol.
 - e. Select **Copy member ID** along with the ID.
 - f. Paste this member ID in the **User ID** field in the **Create Contact Point** dialog box.
 - g. Click **OK**.

When the sales representative now executes a slash command in Slack, his resource record in Fusion Sales is identified based on his Slack user ID. And the query is executed on his behalf. The information from his record in Fusion Sales is then displayed in Slack.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Set Up a Scheduled Process to Synchronize Slack Users

This is a required step for Service from "How You Set Up a Slack Channel for Your Fusion Application".

(Required for your service application, Not Required for your sales application)

To set up a scheduled process to synchronize Slack users:

1. In the Navigator for your Fusion application, click **Scheduled Processes**.
2. In the Overview page, click **Schedule New Process**.
3. Specify the name as **Sync All Slack Resources**.
4. Click **Advanced**.
5. Schedule the job to execute on a recurring basis.

You have now completed the setup of the Slack channel for collaboration. While composing messages in the Messages tab of an SR, all the Slack channels and users for a corresponding workspace appear in the **To** drop-down list.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Set Up a Slack Channel: Method 2

How do I set up a Slack channel: Method 2?

How you integrate with Slack depends upon your company's requirement. To create your own slash commands in Slack, follow the steps listed in the following table:

Step no.	Name of topic
1	Create a Slack App for Fusion Sales and Service
2	Update the Display Information for Your Slack App
3	Add a Redirect URL for Your Slack App
4	Add Scopes to Your Slack App
5	Set Up Interactive Components for Your Slack App
6	Set Up Slash Commands in Slack for Opportunities, Leads, and Accounts (Required for your Sales application, Not Required for your Service application)
7	Update Functional Setup Manager with Credentials of Your Slack App
8	Create a Slack Channel in Your Fusion Application
9	Enable Notifications to Slack
10	Add a Slack Contact Point for Your Sales Representatives Who Use Slack (Required for your Sales application, Not Required for your Service application)
11	Set Up a Scheduled Process to Synchronize Slack Users (Required for your Service application, Not Required for your Sales application)

Create a Slack App for Fusion Sales and Service

To create a Slack app for Oracle Fusion Sales and Service:

1. Navigate to: `https://api.slack.com/apps`
2. Click **Create an App**.
3. In the **Create a Slack App** dialog box, in the **App Name** field, type a suitable app name.
4. From the **Development Slack Workspace** drop-down list, select the workspace where you want the Fusion Sales and Fusion Service integration to be deployed.
5. Click **Create App**.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Update the Display Information for Your Slack App

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

To update the display information for your Slack app:

1. Navigate to the Settings region of your Slack app.
2. Click **Basic Information**.
3. On the Basic Information page, scroll down to the Display Information region.

The name of your Slack app is prefilled in the **App Name** field.

4. Provide a short description, app icon, and background color for your Slack app.
5. Click **Save Changes**.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

How do I add a redirect URL in my Slack app?

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

After a Slack channel is authorized, the redirect URL is used by Slack to provide OAuth tokens to Fusion Sales and Fusion Service.

To add a redirect URL for your Slack App:

1. In the Features region of your Slack app, click **OAuth & Permissions**.
2. On the **OAuth & Permissions** page, in the Redirect URLs region, click **Add New Redirect URL**.
3. In the **Redirect URLs** field, set the redirect URL.

Note: The URL must be in the following format: `https://pod_name:443/crmUI/faces/AuthenticateSlack`, where `pod_name` is the name of the Fusion Sales and Fusion Service instance provisioned for you.

4. Click **Add**.
5. Click **Save URLs**.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Add Scopes to Your Slack App

The capabilities and permissions of a Slack app are governed by the scopes that it requests.

Note: For more information about Slack permission scopes see the Slack API documentation: [Slack Permission Scopes](#).

To add the scopes to your Slack app:

1. In the Features region of your Slack app, click **OAuth and Permissions**.
2. On the OAuth and Permissions page, navigate to the Scopes region.
3. In the Bot Token Scopes section, click **Add an OAuth Scope**.
4. Add the following scopes to your app:

- `channels:manage`
- `channels:read`
- `chat:write`
- `chat:write.customize`
- `chat:write.public`
- `Commands`
- `groups:read`
- `groups:write`
- `im:write`
- `im:read`
- `mpim:read`
- `mpim:write`
- `users:read`

Note: A bot user is created in your app.

5. (Optional) Rename the bot user by navigating to the App Home region of your Slack app.

Set Up Interactive Components for Your Slack App

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

Interactive components are action buttons that are displayed on Slack messages. Users can take these actions on the messages in Slack, which are sent from Fusion Sales and Service.

For example: When an SR is forwarded to a Slack channel, users can click Reply to reply to an SR. They can then add their response in the resulting dialog.

To set up interactive components for your Slack app:

1. In the Features region of your Slack app, click **Interactivity and Shortcuts**.
2. In the Interactivity region, turn on the interactivity.
3. In the **Request URL** field, set the request URL to `https://pod_name:443/crmRestApi/collaborationResources/latest/actions`, where `pod_name` is the name of the Oracle Fusion Sales and Service instance provisioned for you.
4. Click **Save Changes**.

Related Topics

- [Use the Slack Channel for Collaboration](#)

Set Up Slash Commands in Slack for Opportunities, Leads, and Accounts

This is a required step for Sales from "How You Set Up a Slack Channel for Your Fusion Application".

(Required for your sales application, Not Required for your service application)

Because sales representatives travel quite often, sometimes it's not easy for them to sign in to Fusion Sales on their different devices. With the integration of Fusion Sales with Slack, they can view their top five opportunities, leads, and accounts from Slack on any of their devices. They can also take actions on their opportunities.

They can navigate to a workspace in Slack and use the respective slash commands to get their top five opportunities, leads, and accounts. The slash commands use the Sales Cloud RESP API to retrieve the opportunities, leads, and accounts. The results are only visible to the Slack user who executes the command. You first need to set up the slash commands for opportunities, leads, and accounts in Slack.

Here's a list of slash commands in Slack and their corresponding results.

Slash Command in Slack	What this command retrieves from Fusion Sales
<code>/ec-my-opportunities</code>	Top five opportunities with the close date in the current quarter and owned by the sales representative who executes the command.
<code>/ec-my-leads</code>	First five leads owned by the sales representative who executes the command. They are selected from only the qualified and unqualified leads owned by the sales representative in the last three months.
<code>/ec-my-accounts</code>	First five accounts owned by the sales representative who executes the command.

To set up the slash commands in Slack for opportunities, leads, and accounts:

1. In the Features region of your Slack app, click **Slash Commands**.
2. In the Slash Commands region, click **Create New Command**.
3. In the **Command** field, enter the following commands:
 - For opportunities: `/ec-my-opportunities`

- For leads: `/ec-my-leads`
- For leads: `/ec-my-accounts`

Note: Ensure that the commands exactly match the given commands. If you modify the commands, they don't work.

4. In the **Request URL** field, enter the following URL: `https://pod_name/crmRestApi/collaborationResources/latest/commands`
Here, `pod_name` is the name of the Fusion Sales and Service instance provisioned for you.
5. Provide an appropriate **Short Description**. For example: `Get my first 5 opportunities`
6. You can leave the **Usage Hint** field blank, as additional parameters aren't supported.
7. Click **Save** to save the command or click **Cancel** to discard it.

Limitations for Activity Type When a Slash Command Is Used

For each opportunity returned by the `/ec-my-opportunities` slash command, the user can create a task, log a call, or schedule an appointment. These activities are created on the opportunities. In Fusion Sales and Service, you can add a custom lookup as the parent of the Activity Type lookup to further constrain the list of activity types.

But the Slack integration is unable to constrain the list of values for the activity type based on custom parent lookup type. So the users see an error if they select an activity type that isn't available for the activity they're creating.

Update Functional Setup Manager with Credentials of Your Slack App

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

To update the Functional Setup Manager (FSM) in the Fusion application with the credentials of your Slack app:

1. In the Settings region of your Slack app, click **Basic Information**.
2. From the App Credentials section of the Basic Information page, copy the values of the **Client ID**, **Client Secret**, and **Signing Secret**.
3. Sign in to your Fusion application as a setup user or administrator.
4. Navigate to the **Setup and Maintenance** work area and open the **Tasks** panel tab.
5. Click **Search** from the list of displayed tasks.
6. Search for and select **Manage Collaboration Network Credentials**.
7. In the Manage Collaboration Network Credentials page, specify the **Client Identifier**, **Client Secret**, and **Signing Secret** that you copied from your Slack app.
8. Click **Save and Close**.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Create a Slack Channel in Your Fusion Application

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

For each of your Slack workspaces, you must create a Slack channel in Fusion Sales and Service. So if you have multiple Slack workspaces, you have multiple Slack channels in Fusion Sales and Service.

To create a Slack channel in the Fusion application:

1. Sign in to the Fusion application 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. Search for and select **Manage Communication Channels**.
5. In the Service Channels page, click **Create Channel**.
6. In the Create Channel window:
 - The **Stripe Code** is set as **CRM** by default.
 - Select the **Purpose** as **Collaboration**.
 - Select the **Channel Type** as **Slack**.
 - Click **Authorize**.

A new browser tab opens and the following page is displayed in Slack: Sign in to your workspace.

7. In the space provided below the **Enter your workspace's Slack URL** prompt, enter the workspace for which you want to create the Slack channel.
8. Click **Continue**.
9. Sign in to your Slack workspace as an administrator.
10. Complete the next steps on Slack to provide permissions.
11. Click **Authorize**.

A message is displayed, stating that the authorization is complete.

12. Close this browser tab and return to the channel creation page in Fusion Sales and Fusion Service.
Notice that the **Account Name**, **Channel Code**, and **Display Name** are prepopulated.
13. Click **Save and Close** to save the Slack channel.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Enable Notifications to Slack

This is a required step from "How You Set Up a Slack Channel for Your Fusion Application".

All outbound messages from Fusion Sales and Fusion Service to Slack use the notifications framework. So, for outbound messages to be delivered to Slack, you must enable notifications.

For more information about how to enable notifications, see "Enable Notifications for Service Requests".

Related Topics

- [How do I enable notifications?](#)
- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Add a Slack Contact Point for Your Sales Representatives Who Use Slack

This is a required step for Sales from "How You Set Up a Slack Channel for Your Fusion Application".

(Required for your sales application, Not Required for your service application)

Every Slack user has a unique Slack user ID. When a sales representative executes a slash command in Slack, his information needs to be retrieved from Fusion Sales and displayed in Slack. For this to happen, you must add a Slack contact point in Fusion Sales for every sales representative who uses Slack. This way, you map the unique Slack user ID of each sales representative to his resource record in Fusion Sales.

1. In the Navigator for Fusion Sales, click **Resource Directory**.
2. Click the **Tasks** panel tab.
3. In the **Resources** region, click **View Resources**.
4. On the **View Resources** page, navigate to the **Resource Name** field.
5. Type the name of the resource to whom you want to give the capability to execute Slash commands.
6. Click **Search**.
7. In the **Search Results** area, click the record of the sales representative.
8. On the **Resource** page of the sales representative, click the **Additional Contact Information** tab.
9. Click the plus icon to create a new contact point.
10. In the **Create Contact Point** dialog box, in the **Contact Point Type** drop-down list, select **Social network**.
11. In the **Purpose** field, select **Work**.
12. In the **Social Network** drop-down list, select **Slack**.
13. In the **User ID** field, find and enter the Slack user ID of the service representative as follows:
 - a. In the Slack workspace, in the **Direct Messages** area, select the name of the sales representative.
 - b. In the adjacent pane, click the name of the sales representative.
 - c. Click **View profile**.
 - d. In the **Workspace Directory** panel that's displayed, click the ellipsis symbol.
 - e. Select **Copy member ID** along with the ID.
 - f. Paste this member ID in the **User ID** field in the **Create Contact Point** dialog box.
 - g. Click **OK**.

When the sales representative now executes a slash command in Slack, his resource record in Fusion Sales is identified based on his Slack user ID. And the query is executed on his behalf. The information from his record in Fusion Sales is then displayed in Slack.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Set Up a Scheduled Process to Synchronize Slack Users

This is a required step for Service from "How You Set Up a Slack Channel for Your Fusion Application".

(Required for your service application, Not Required for your sales application)

To set up a scheduled process to synchronize Slack users:

1. In the Navigator for your Fusion application, click **Scheduled Processes**.
2. In the Overview page, click **Schedule New Process**.
3. Specify the name as **Sync All Slack Resources**.
4. Click **Advanced**.
5. Schedule the job to execute on a recurring basis.

You have now completed the setup of the Slack channel for collaboration. While composing messages in the Messages tab of an SR, all the Slack channels and users for a corresponding workspace appear in the **To** drop-down list.

Related Topics

- [How do I set up a Slack channel: Method 2?](#)
- [Use the Slack Channel for Collaboration](#)

Slack Notifications

Send Notifications to Slack Using Groovy Triggers or Object Workflow

Notifications can be sent from Fusion Sales and Service to Slack by using groovy triggers or object workflow.

To understand how you can use groovy triggers to send notifications to Slack, see the following topics:

- [Define a Groovy Trigger to Send Notifications to Slack](#)
- [Examples of Groovy Scripts for Sending Notifications to Slack](#)

To send notifications to Slack by using object workflow, you must create the object workflow and select the Groovy Script action. For more information and examples, see "Object Workflows and Groovy Scripts: How They Work Together".

Related Topics

- [Define a Groovy Trigger to Send Notifications to Slack](#)
- [Examples of Groovy Scripts for Sending Notifications to Slack](#)
- [How Object Workflows and Groovy Scripts Work Together](#)

Define a Groovy Trigger to Send Notifications to Slack

Before writing the groovy scripts, ensure that you complete these steps:

- Create a Slack Channel. For more information, see "Create a Slack Channel in Your Fusion Application".
- Enable notifications for service requests. For more information, see "Enable Notifications for Service Requests".

To define a groovy trigger to send notifications to Slack:

1. Sign in to the application as an administrator.
2. Ensure that you're in a sandbox.
3. Navigate to Application Composer.
4. In the **Application** field, select **CRM Cloud** from the drop-down list.
5. In the Object Tags region, select the **Service** check box.
Note: If you're writing a groovy trigger on a Sales object, select the **Sales** check box.
6. Expand the **Service Request** object.
7. Click **Server Scripts**.
8. Select the Triggers tab.
9. Create an object trigger for Service Request Server Scripts.
10. Define your groovy trigger as described in "Examples of Groovy Scripts for Sending Notifications to Slack".

Related Topics

- [Create a Slack Channel in Your Fusion Application](#)
- [How do I enable notifications?](#)
- [Examples of Groovy Scripts for Sending Notifications to Slack](#)

Examples of Groovy Scripts for Sending Notifications to Slack

Here are some examples of groovy scripts for sending notifications from Fusion Service to Slack:

Trigger Example: SR Critical Check Box Change

Use the following code to trigger a notification to be sent to Slack when the Critical check box on the service request (SR) is updated:

Note: All the steps to set the HashMap in these examples are required to send notifications to Slack. Currently, Slack notifications don't use the Notification Preferences Manager. So it's necessary to pass the HashMap to the sendNotification method.

```
// TRIGGER TYPE: AFTER CHANGES POSTED TO DATABASE
// Notify #support channel in OraSvc workspace when the Critical check box in an SR is selected or
// deselected
int indexOfCF = adf.oldValue.getAttributeIndex('CriticalFlag')
def oldValue = adf.oldValue.getAttribute(indexOfCF,oracle.jbo.server.EntityImpl.TRANS_ORIGINAL_VERSION)
def newValue = getAttribute('CriticalFlag')

if (oldValue != newValue) {
    println("Triggering notification...") // For debugging purpose only
    try {
        def map = new HashMap()
        map.put("Channels",["ORA_SVC_SLACK"])
        map.put("MessageText","Test Message for a change to the Critical check box on the SR")
        map.put("ChannelAccountName", "A1B2C3D4F5") // This is a very important step to send notifications to
        // Slack. Pass the Account Name field for the Slack channel that you created
        map.put("SlackChannels",["@kanyuev", "#general", ...]) // This is also an important step. You can provide
        // one or more Slack channels or users here

        adf.util.sendNotification(adf, map)
```



```
println("Success.") // For debugging purpose only
} catch (e) {
    throw new oracle.jbo.ValidationException('Failure: ' + e.getMessage())
}
}
```

Trigger Example: SR Created

Use the following code to trigger a notification to be sent to Slack when an SR is created:

```
// TRIGGER TYPE: BEFORE INSERT TO DATABASE
// Notify #support channel in OraSvc workspace when an SR is created
try {
    def map = new HashMap()
    map.put("Channels", ["ORA_SVC_SLACK"])
    map.put("MessageText", "An SR is created")
    map.put("ChannelAccountName", "A1B2C3D4F5") // This is a very important step to send notifications to
    // Slack. Pass the Account Name field for the Slack channel that you created
    map.put("SlackChannels", ["@kanyuev", "#general", ...]) // This is also an important step. You can provide
    // one or more Slack channels or users here

    adf.util.sendNotification(adf, map)
} catch (e) {
    throw new oracle.jbo.ValidationException('Failure: ' + e.getMessage())
}
```

Related Topics

- [Define a Groovy Trigger to Send Notifications to Slack](#)

Validate Whether Notifications Are Sent to Slack

You can validate if the notifications are sent to Slack by updating a service request (SR) in Oracle Fusion Service.

Note: Notifications to Slack can be sent for other objects as well, such as opportunity and custom objects.

Validation Example: When You Use the Critical SR Check Box Change Trigger

If you use the Critical SR Check Box Change trigger, here's how you validate if a notification is sent from Oracle Fusion Service to Slack:

1. Sign in to Fusion Service.
2. Create and save an SR.
3. In the SR Summary tab, select the **Critical** check box and save the SR.
4. Verify whether you have received a notification in Slack.

User Credentials for Executing Commands

Set Up User Credentials for Executing Commands from a Collaboration Network

To execute the Oracle Fusion Sales and Service REST API commands from a collaboration network such as Slack, you can use a predefined Application ID user. The predefined Application ID user has many privileges.

You also have the option to set up the user credentials instead of using a predefined user. The user that you set up can have only those privileges required to execute the commands from a collaboration network such as Slack.

To set up the user credentials to execute the commands from a collaboration network such as Slack:

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. Search for and select the Manage Collaboration Network Credentials task.
4. On the Manage Collaboration Network Credentials page, navigate to the Fusion Sales and Service Application Credentials region.
5. Enter the appropriate user ID and password.

FAQs About Slack Integration

Frequently Asked Questions (FAQs) for Slack Integration

This section contains some frequently asked questions (FAQs) when setting up and maintaining slack integration. Scroll or search to find answers to common questions.

Why is there a banner recommending I update the permission scopes for my Slack app?

When you're editing a Slack app, you may see a banner recommending you to update the permission scopes for your app. If you see this banner, then it means that you're using the classic version of the app, which uses a coarser permission scope.

The latest version uses fine-grained permission scopes, and you can request only the specific scopes required by your app. If you prefer to continue using the coarser scope, you don't have to do anything. But if you prefer to use the fine-grained permission scopes, then here's what you must do:

1. On the Slack website, navigate to the page where you manage your applications.

Note: You must be an administrator of your Slack application.

2. You should see a banner indicating that you can start using a finer-grained scope.
3. Click the link in the banner and follow the steps to upgrade your Slack app.
4. In your Fusion application, navigate to the **Setup and Maintenance** work area.
5. Open the **Tasks** panel tab.
6. Click Search from the list of displayed tasks.
7. Search for and select the **Manage Communication Channels** task.
8. Reauthorize your Slack channel.

After the channel is authorized, the integration starts using the finer scope.

Related Topics

- [Add Scopes to Your Slack App](#)

24 Collaborate with Slack

How do I collaborate with Slack?

Using Slack, your service agents can securely collaborate with subject matter experts to quickly resolve customer questions and issues.

You can integrate your sales and service application with Slack using two different methods. These methods are described in detail in the Slack Channels chapter of the Oracle Fusion Service Implementing Service guide.

Oracle supports both public facing and workspace private uses of Slack.

For Slack security layers, Oracle supports `https` respecting Oracle Fusion and Slack signing key secret, including Oracle Fusion data security using `pubs` framework (a proxy user framework). The credentials are stored in a credential store. No one has access except the application, but an administrator can configure it.

Slash commands can be configured. There's a configuration page where you can configure slash commands.

APIs are managed through an admin page that interacts with WLD credentials store.

Note: The Send Additional SR Fields to Slack section isn't applicable for Redwood Service Center. So you can ignore the following topic in that section: Send Additional Fields When an SR Is Forwarded to Slack.

Related Topics

- [How do I set up a Slack channel?](#)
- [How do I set up a Slack channel: Method 2?](#)

Create a Collaboration Action for Slack

Using the Administration UI, you can set up or create different collaboration actions.

Let's see how to set up a collaboration action to start an internal conversation in a service request (SR) with a user in Slack. You can also define which attributes of the business object can be shared with Slack as part of the internal conversation.

Note: This example is for starting an internal conversation in an SR. You can also use any other suitable action names different from those given in the examples. For this example:

- A child action would be to create a **Reply to Service Request** button. It's a child of the first message received in Slack from Service Center.
- Further, a child of the **Reply to Service Request** child action would be to create a **Submit Reply** action. When a Slack user enters a message and clicks **Submit Reply**, the message goes to Service Center.

These two child actions have been described in the two subsequent topics.

To set up a collaboration action using the Administration UI:

1. Sign in as an administrator.
2. Navigate to the **Setup and Maintenance** work area and open the **Tasks** panel tab.
3. Search for and click the **Manage Collaboration Actions** task.

The All Collaboration Actions page opens in a new tab.

4. Click **Create Collaboration Action**.
5. In the Action Details region, specify the following details:
 - a. **Action Name:** Specify a name. For example: Start Internal Conversation in SR.
 - b. The **Action** field is automatically populated.
 - c. **Description:** Add a suitable description.
 - d. **Action Code:** Specify an action code of your choice. For example: `start-collab-on-SR`.
 - e. **Business Object:** Select the business object on which you're creating the collaboration action. For example, Service Request, Opportunity, Lead, and so on.
 - f. **Action Type:** From this drop-down list, select **Start a Collaboration Conversation**.

Note: For this Start a Collaboration Conversation action, the following fields aren't required: **Action Method**, **Display Sequence**, and **User Interface Component**.

6. Click **Next**.
7. In the Request Details region, specify the following details:
 - a. **Request Content Type:** This field is automatically populated.
 - b. **Request End Point:** Enter the following end point: `/crmRestApi/resources/latest/serviceRequests?q=SrId=[[Srid]]`

The `/crmRestApi/resources/latest/serviceRequests` part is automatically populated in the field. To facilitate querying the SR object using the primary key field, you're appending the following at the end: `?q*Srid*[[Srid]]`
 - c. **Request Payload:** You can skip this field.
8. Click **Next**.
9. In the Response Details region, specify the following details:
 - a. **Response Visibility:** From this drop-down list, select **Public**.
 - b. **Response Type:** From this drop-down list, select **Action Attributes**.
 - c. **Response Content Type:** This field is automatically populated.
 - d. **Response Payload:** You can skip this field.
10. Click **Next**.
11. In the Add Action Attribute region, you must add some SR attributes that you would like to share with Slack users when starting an internal conversation.

- a. Click **Add Attribute**.
- b. In the **Add Action Attribute** dialog box, enter the following details:

Note: See the table at the end of this step for some specific details that you must set for some attributes.

- **Attribute Name:** Select a name from the drop-down list.
- **Display Name:** Specify a suitable display name.
- **Data Type:** Select the appropriate value from the drop-down list.

- **Default Value:** Specify a default value.
- **User Interface Component:** This field can have the following values: List, Text Area, or Text Box. The default value is Text Box.
- **Display Sequence:** Specify a high or low number depending on where you want the UI component to be displayed. For example: You can specify a higher number such as 12 if you want it to be displayed lower down on the page.
- **Business Object Identifier:** If it's a field that's used for identifying a record, then you must select this switch.

Note: Let's say a to and fro message thread about a particular SR exists between Slack and Service Center. Then the linking of the messages is done through this field. You can select SR ID or SR number as the Business Object Identifier, because they both uniquely identify as an SR.

- **Included in Request:** If information is being sent from Slack to Service Center, then select this switch.
- **Included in Response:** If you're sending a response message from Service Center to Slack, then select this switch.
- **End Point:** This field is applicable only if you select the User Interface Component as List.
- **Option Label:** This field is applicable only if you select the User Interface Component as List.
- **Option Value:** This field is applicable only if you select the User Interface Component as List.
- **Deep Link:** If required, provide a deep link that uniquely identifies the SR. For example: Link to SR number that shows up in the Slack UI: `crmUI/service/vp/ec/container/sr/edit?srNumber=[ [SrNumber] ]`. If you want one of the other attributes to be a link, you can add the deep link to the SR by editing that attribute. For example, you can add the deep link to the SR for the title attribute.

Here's a list of some specific details that you must set for the following common action attributes:

Action Attribute	Required Details
SrId	Select the Business Object Identifier switch.
SrNumber	Select the Include in Response switch.
SrNumber	Provide the following deep link in the Deep Link field: <code>crmUI/service/vp/ec/container/sr/edit?srNumber=[[SrNumber]]</code>
Title	Select the Included in Response switch.
Problem Description	Select the Included in Response switch.

12. Click **Save**.

Create a Child Action for a Collaboration Action for Slack

A child action is an action within the main collaboration action.

For example: For the Collaboration on Service Request action, you would create a **Reply to Service Request** button as the child. The **Reply to Service Request** button is a child of the first message received in Slack from Redwood Service Center about needing help with an SR. Along with the message, some basic information about the SR is shared.

Note: You can use any other suitable action name instead of **Reply to Service Request**. Whatever action name you use is displayed in the Slack UI.

To create a child action for a collaboration action:

1. Sign in as an administrator.
2. Navigate to the **Setup and Maintenance** work area and select the Service offering.
3. Open the **Tasks** panel tab.
4. Search for and click the **Manage Collaboration Actions** task.
The All Collaboration Actions page opens in a new tab.
5. From the displayed list of existing collaboration actions, click the collaboration action for which you want to create a child action.
The collaboration action opens, and its name is displayed in the navigation pane.
6. Click the plus symbol next to the name of the collaboration action.
7. Specify all the details for the child action as explained in Create a Collaboration Action. Considering the example of the **Reply to Service Request** button, you would specify the following details:
 - a. Action Details region:
 - **Action Name:** Reply to Service Request
 - **Action Code:** Specify a value of your choice. For example: reply-to-sr
 - **Business Object:** Select **Conversation Message** from the drop-down list.
That's because we're creating a conversation message under the request.
 - **Action Type:** Select **Open a Dialog** from the drop-down list.
 - b. Request Details region:
The Request End Point is populated automatically, so you don't have to change anything here.
 - c. Response Details region:
You can leave this page blank. That's because clicking the **Reply to Service Request** button would show a message text box in Slack. And you're not sending a response message from Redwood Service Center to Slack.
 - d. Add Action Attribute region:
Here, you would add only one attribute, which is **Message Content**.
 - **Attribute Name:** Select **MessageContent** from the drop-down list.

- **Data Type:** Select **String** from the drop-down list.
- **User Interface Component:** Select **Text Area** from the drop-down list.
- Select the **Included in Request** switch.

8. Click **Save**.

Note: Instead of clicking the **Reply to Service Request** button, the subject matter expert (SME) can also reply by clicking the native **Reply** action. The SME must mention the app in the format @AppName in the native Reply. Here, AppName is the name of your company's Slack app created for Oracle Fusion Service. When the SME mentions your app in this format, your app adds their reply as a message on the request.

Create a Second-Level Child Action for a Collaboration Action for Slack

The child of the Reply to Service Request action in Slack would be the Submit Reply action. When a Slack user enters a message and clicks Submit Reply, the message goes to Redwood Service Center.

Note: You can use any other suitable action name instead of **Submit Reply**. Whatever action name you use is displayed in the Slack interface.

To create the **Submit Reply** child action to the **Reply to Service Request** action:

1. Specify all the details for the child action as explained in Create a Collaboration Action. Considering the example of the **Reply to Service Request** button in the previous topic, you would specify the following details:

a. Action Details region:

- **Action Name:** Submit Reply
- **Action Code:** You can type any value of your choice here.
- **Business Object:** Select **Conversation Message** from the drop-down list.
- **Action Type:** Select **Call an API** from the drop-down list because it's calling a Redwood Service Center API.
- **Action Method:** Select **Post** from the drop-down list because you're creating a new message.
- **Display Sequence:** Select 1.

b. Request Details region:

- **Request Content Type:** This field is populated automatically.
- **Request End Point:** This field is populated automatically.
- **Request Payload:** Add the following payload in this field:

```
{ "MessageContent": "[[MessageContent]]", "MessageTypeCd":  
"ORA_SVC_INBOUND", "ParentMessageId": [[ConversationMessageId]], "SenderAlternateId":  
[[ $system.CollabResourceId]], "SenderFreeFormAddress":
```

```
"[[${system.ExternalResourceDisplayName}]]", "MessageAlternateId":  
"[[${system.ExternalMessageId}]]", "recipients": [[${system.ConversationMessageRecipients}]]}
```

c. Response Details region:

- **Response Visibility:** Select Public from the drop-down list.
- **Response Payload:** Set the value to: `{"text": "[[MessageContent]]"}`

This message content is also echoed to all the other SMEs in Slack, so they all can view the reply given by the first SME. And they can add on to it if required.

- **Response Type:** Select **User-Defined JSON Formatted Result** from the drop-down list.
- **Response Content Type:** This field is populated automatically.

d. Add Action Attribute region:

This page has only one attribute, which is **Message Content** because that's the only attribute for Submit Reply.

- Ensure that you select the **Included in Request** switch.

2. Click **Save**.

Add an Event to Capture Messages Mentioning Your Slack App

Let's say an agent needs help with resolving a service request (SR), and they post a question on Slack. A subject matter expert (SME) can reply to their question by clicking either the **Reply to Service Request** button or the native **Reply** action.

When the SME clicks the native **Reply**, they must mention the app in the format @AppName in the native Reply. Here, AppName is the name of your company's Slack app created for Redwood Service Center. When the SME mentions your app in this format, their reply is captured as a message on the SR. But for this to happen, you must add an event in your Slack app to capture such replies.

To add an event that captures messages mentioning your Slack app as conversation messages in Redwood Service Center:

1. Open your Oracle app in Slack.
2. In the **Features** section of the navigation pane, click **Event Subscriptions**.
3. On the Event Subscriptions page, in the **Enable Events** area, turn on the switch to enable events.
4. In the **Request URL** field, type the following:

```
https://{URL}/crmRestApi/collaborationResources/latest/slackAppMention
```

Note: Remember to replace {URL} in this link with the base URL of your pod.

5. In the **Subscribe to bot events** area, click **Add Bot User Event**.
6. Click **app_mention**.

The app_mention event name is displayed in the Subscribe to bot events area, along with the description and scope.

7. Click **Save Changes**.

25 Collaborate with Microsoft Teams

Prerequisites for Setting Up a Microsoft Teams Channel for Fusion Service

Before you set up the integration of your service application with Microsoft Teams, you must ensure that you have the following:

- The appropriate access or subscription to Microsoft Teams that permits you to use Developer Portal to create an app for Fusion Service.
- Administrator privileges that let you to sign in to the Azure portal.

Related Topics

- [How do I Set Up a Microsoft Teams Channel?](#)

How do I Set Up a Microsoft Teams Channel?

Using Microsoft Teams, your service agents can collaborate with subject matter experts to resolve customer questions and issues faster. Here's the sequence of steps that you must complete to set up the integration of your service application with Microsoft Teams:

Note: Each of these steps is described in a separate topic, and you must complete them in the given order.

Step no.	Name of topic
1	Create a Microsoft Teams App for Service Center
2	Set Up a Bot User for Your Microsoft Teams App
3	Register Your Microsoft Teams App in the Azure Portal
4	Enable Collaboration With Microsoft Teams in FSM
5	Update FSM with Credentials of Your Microsoft Teams App
6	Create a Microsoft Teams Channel in Redwood Service Center
7	Enable Notifications to Microsoft Teams

Step no.	Name of topic
8	Create a Collaboration Action
9	Create a Child Action for a Collaboration Action
10	Create a Child Action for the Reply to Service Request Child Action
11	Configure the FND_IDP_PROXY_USER_WHITELIST Profile Option
12	Upload Your Microsoft Teams App
13	Install Your Microsoft Teams App and Add It to Your Team

Create a Microsoft Teams App

To create a Microsoft Teams app for Redwood Service Center:

1. In Microsoft Teams, click **Apps** in the navigation pane.
2. On the Apps page, search for and add **Developer Portal**.
3. In the **Manifest editor** tab of **Developer Portal**, click **Create a new app**.
4. On the **App details** page, specify the details in the following sections:
 - a. **App names:** Enter the short and full names for your app. The short name that you specify automatically appears in the navigation pane.
 - b. **Identification:** Remember to click **Generate** below the **App ID** field to generate the app ID.
 - c. **Descriptions:** Enter the short and full descriptions for your app.
 - d. **Developer information:** Enter your company name and website.
 - e. **Partner information:** If a partner organization is building the app for you, you must enter their Microsoft Partner Network ID. This section is optional and isn't applicable if your company is building the app.
 - f. **App URLs:** Provide the links to your policy statement and terms of use.
 - g. **Branding:** Upload the two icons for your app as required, and follow the instructions to select an app accent color.
5. Go to the **Capabilities** section of the navigation pane to set up a bot user for your Microsoft Teams app.

For more details, see Set Up a Bot User for Your Microsoft Teams App.
6. Go to the **Finish** section of the navigation pane and click **Test and distribute**.
7. Click **Download**.

The Microsoft Teams app that you created for Redwood Service Center is downloaded as a compressed file.

Set Up a Bot User for Your Microsoft Teams App

To set up a bot user for your Microsoft Teams app:

1. In Microsoft Teams, go to the **Capabilities** section of the navigation pane.
2. Click **Bots**.
3. The **Set up a bot** dialog box displays two tabs:
 - o New bot: Click this tab to create a new bot user for your app.
 - i. In the **Name** field, enter a name for your new bot.
 - ii. In the **Messaging bot** and **Calling bot** regions, select the check boxes that are relevant for your company.
 - iii. In the **Scope** region, select the **Personal**, **Team**, and **Group Chat** check boxes.
 - iv. Click **Create bot**.
 - o Existing bot: Click this tab to configure an existing bot user for your app.
 - i. In the **Bot** dialog box, in the **Bot ID** region: You can either select from an existing bot or connect to a different bot ID by specifying the ID.
 - ii. In the **Messaging bot** and **Calling bot** regions, select the check boxes that are relevant for your company.
 - iii. In the **Scope** region, select the **Personal**, **Team**, and **Group Chat** check boxes.
 - iv. Click **Save**.
4. On the **Bots** page, in the **App passwords** region, click **Generate new password**.

Note: You must save this generated password. This password is used later as the client secret when you configure the MS Teams credentials in Service Center.

5. In the **Messaging endpoint** region, enter the bot endpoint address. For example: `https://environment-name/crmRestApi/collaborationResources/latest/activities`

Here, environment-name refers to the environment where you have deployed Oracle Service Center.

Register Your Microsoft Teams App in the Azure Portal

Ensure that you have an Azure account before you register your Microsoft Teams app in the Azure portal.

To register your app in the Azure portal:

1. Go to the Azure Portal at: `portal.azure.com`
2. Navigate to the Manage Azure Active Directory region and click **View**.
3. In the navigation pane, click **App registrations**.

In the App registrations page, you can see the new Microsoft Teams app that you created.
4. Click your app.
5. Copy the following information from the **Essentials** region:

- **Application (client) ID:** You must enter this application client ID while updating FSM with the credentials of your Microsoft Teams app.
- **Directory (tenant) ID:** You use this ID as the account name of the Microsoft Teams channel that you create in Fusion Service.

Enable Collaboration With Microsoft Teams in Functional Setup Manager

You can enable collaboration of Service Center with Microsoft Teams by turning on the administration UI.

After the administration UI is turned on, you can see all the tasks related to Microsoft Teams integration in Functional Setup Manager (FSM). For example: Manage Microsoft Teams Credentials, Manage Collaboration Network Credentials, and Manage Collaboration Actions. The Manage Collaboration Actions task takes you to all the actions in the administration UI.

To enable collaboration of Service Center with Microsoft Teams in FSM:

1. Sign in as an administrator or a setup user.
2. Navigate to the **Setup and Maintenance** work area and select the **Service** offering.
3. Select the **Communication Channels** functional area.
4. Click the **Change Feature Opt In** link.
5. On the Opt In: Service page, select the **Productivity Tools** row.
6. Ensure that the **Enable** check box is selected and click **Edit** in the **Features** column.
7. On the Edit Features: Productivity Tools page, navigate to the Collaborate using External Networks region.
8. For **Microsoft Teams**, select the **Enable** check box.
9. Click **Done**.

Update Functional Setup Manager with Credentials of Your Microsoft Teams App

To update the Functional Setup Manager (FSM) in Fusion Service with the credentials of your Microsoft Teams app:

1. Sign in to Service Center as a setup user or administrator.
2. In the Setup and Maintenance work area, go to the **Tasks** panel tab.
3. Search for and click **Manage Microsoft Teams Credentials**.
The Manage Collaboration Network Credentials page is displayed.
4. In the Collaboration Network Application Credentials region, enter the required information in the following fields:
 - **Client ID:** Enter the Application Client ID that you copied from the Azure portal. For more information, see Register Your Microsoft Teams App in the Azure Portal.
 - **Client Secret:** Enter the password that you generated in the **App passwords** region of the **Bots** page. For more information, see step 3 in Set Up a Bot User for Your Microsoft Teams App.
5. Click Save and Close.

Create a Microsoft Teams Channel

To create a Microsoft Teams channel in Redwood Service Center:

1. Sign in as an administrator or a setup user.
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. Search for and select **Manage Communication Channels**.
5. In the Service Channels page, click **Create Channel**.
6. In the **Create Channel** window, select **CRM** in the **Stripe Code** field.
7. Select the **Purpose** as **Collaboration**.
8. Select the **Channel Type** as **Microsoft Teams**.
9. In the **Account Name** field, enter the directory tenant ID that you copied from the Azure portal.

For more information, see [Register Your Microsoft Teams App in the Azure Portal](#).

10. The **Channel Code** field is autogenerated.

The channel code uniquely identifies a communication channel when exporting or importing channels from one environment to another.

- If the autogenerated channel code is unique, you can leave it unchanged.
- If the autogenerated channel code isn't unique, add a set of characters to the code to make it unique.

11. Enter a **Display Name** for the Microsoft Teams channel.
12. Select the **Business Unit** from the drop-down list.

Note: You can see this field only if you have enabled multiple business units for your deployment.

13. Select the **Active** check box.
14. Click **Save and Close**.

Related Topics

- [Register Your Microsoft Teams App in the Azure Portal](#)

Send and Receive Attachments from Microsoft Teams Channel Users

You can send and receive attachments when communicating with users in a Microsoft Teams channel.

To send a file as an attachment to a specific Microsoft Teams channel, use the APIs listed under: [List channel messages](#) > [Permissions](#):

- ChannelMessage.Read.All
- ChannelMessage.Read.Group (uses resource-specific consent)

- ChannelMessage.Read.All

Enable Notifications to Microsoft Teams

All outbound messages from Redwood Service Center to Microsoft Teams use the notifications framework. So, for outbound messages to be delivered to Microsoft Teams, you must enable notifications.

For more information about how to enable notifications, see "Enable Notifications for Service Requests".

Related Topics

- [Enable Notifications for Service Requests](#)

Create a Collaboration Action

Using the Administration UI, you can set up or create different collaboration actions.

Let's see how to set up a collaboration action to start an internal conversation in an SR with a user in Microsoft Teams. You can also define which attributes of the business object can be shared with Microsoft Teams as part of the internal conversation.

Note: This example is for starting an internal conversation in an SR. You can also use any other suitable action names different from those given in the examples. For this example:

- A child action would be to create a **Reply to Service Request** button. It's a child of the first message received in Microsoft Teams from Redwood Service Center.
- Further, a child of the **Reply to Service Request** child action would be to create a **Submit Reply** action. When a Microsoft Teams user enters a message and clicks **Submit Reply**, the message goes to Service Center.

These two child actions have been described in the two subsequent topics.

To set up a collaboration action using the Administration UI:

1. Sign in as an administrator.
2. Navigate to the **Setup and Maintenance** work area and open the **Tasks** panel tab.
3. Search for and click the **Manage Collaboration Actions** task.

The All Collaboration Actions page opens in a new tab.

4. Click **Create Collaboration Action**.
5. In the Action Details region, specify the following details:
 - a. **Action Name:** Specify a name. For example: Start Internal Conversation in SR.
 - b. The **Action** field is automatically populated.
 - c. **Description:** Add a suitable description.
 - d. **Action Code:** Specify an action code of your choice. For example: `start-collab-on-SR`.
 - e. **Business Object:** Select the business object on which you're creating the collaboration action. For example, Service Request, Opportunity, Lead, and so on.

f. Action Type: From this drop-down list, select **Start a Collaboration Conversation**.

Note: For this Start a Collaboration Conversation action, the following fields aren't required: **Action Method**, **Display Sequence**, and **User Interface Component**.

6. Click **Next**.
7. In the Request Details region, specify the following details:
 - a. **Request Content Type:** This field is automatically populated.
 - b. **Request End Point:** Enter the following end point: `/crmRestApi/resources/latest/serviceRequests?q=SrId=[[Srid]]`
The `/crmRestApi/resources/latest/serviceRequests` part is automatically populated in the field. To facilitate querying the SR object using the primary key field, you're appending the following at the end: `?q*Srid=[[Srid]]`
 - c. **Request Payload:** You can skip this field.
8. Click **Next**.
9. In the Response Details region, specify the following details:
 - a. **Response Visibility:** From this drop-down list, select **Public**.
 - b. **Response Type:** From this drop-down list, select **Action Attributes**.
 - c. **Response Content Type:** This field is automatically populated.
 - d. **Response Payload:** You can skip this field.
10. Click **Next**.
11. In the Add Action Attribute region, you must add some SR attributes that you would like to share with Microsoft Teams users when starting an internal conversation.
 - a. Click **Add Attribute**.
 - b. In the **Add Action Attribute** dialog box, enter the following details:

Note: See the table at the end of this step for some specific details that you must set for some attributes.

- **Attribute Name:** Select a name from the drop-down list.
- **Display Name:** Specify a suitable display name.
- **Data Type:** Select the appropriate value from the drop-down list.
- **Default Value:** Specify a default value.
- **User Interface Component:** This field can have the following values: List, Text Area, or Text Box. The default value is Text Box.
- **Display Sequence:** Specify a high or low number depending on where you want the UI component to be displayed. For example: You can specify a higher number such as 12 if you want it to be displayed lower down on the page.
- **Business Object Identifier:** If it's a field that's used for identifying a record, then you must select this switch.

Note: Let's say a to and fro message thread about a particular SR exists between Microsoft Teams and Service Center. Then the linking of the messages is done through this field. You can select SR ID or SR number as the Business Object Identifier, because they both uniquely identify an SR.

- **Included in Request:** If information is being sent from Microsoft Teams to Service Center, then select this switch.

- **Included in Response:** If you're sending a response message from Service Center to Microsoft Teams, then select this switch.
- **End Point:** This field is applicable only if you select the User Interface Component as List.
- **Option Label:** This field is applicable only if you select the User Interface Component as List.
- **Option Value:** This field is applicable only if you select the User Interface Component as List.
- **Deep Link:** If required, provide a deep link that uniquely identifies the SR. For example: Link to SR number that shows up in the Microsoft Teams UI: `crmUI/service/vp/ec/container/sr/edit?srNumber=[{SrNumber}]`. If you want one of the other attributes to be a link, you can add the deep link to the SR by editing that attribute. For example, you can add the deep link to the SR for the title attribute.

Here's a list of some specific details that you must set for the following common action attributes:

Action Attribute	Required Details
SrId	Select the Business Object Identifier switch.
SrNumber	Select the Include in Response switch.
SrNumber	Provide the following deep link in the Deep Link field: <code>crmUI/service/vp/ec/container/sr/edit?srNumber=[{SrNumber}]</code>
Title	Select the Included in Response switch.
Problem Description	Select the Included in Response switch.

12. Click **Save**.

Create a Child Action for a Collaboration Action

A child action is an action within the main collaboration action.

For example: For the Collaboration on Service Request action, you would create a **Reply to Service Request** button as the child. The **Reply to Service Request** button is a child of the first message received in Microsoft Teams from Redwood Service Center about needing help with an SR. Along with the message, some basic information about the SR is shared.

Note: You can use any other suitable action name instead of **Reply to Service Request**. Whatever action name you use is displayed in the Microsoft Teams interface.

To create a child action for a collaboration action:

1. Sign in as an administrator.
2. Navigate to the **Setup and Maintenance** work area and select the Service offering.
3. Open the **Tasks** panel tab.

4. Search for and click the **Manage Collaboration Actions** task.

The All Collaboration Actions page opens in a new tab.

5. From the displayed list of existing collaboration actions, click the action for which you want to create a child action.

The collaboration action opens, and its name is displayed in the navigation pane.

6. Click the plus symbol next to the name of the collaboration action.

7. Specify all the details for the child action as explained in Create a Collaboration Action. Considering the example of the Reply to Service Request button, you would specify the following details:

a. Action Details region:

- **Action Name:** Reply to Service Request
- **Action Code:** Specify a value of your choice. For example: reply-to-sr
- **Business Object:** Select **Conversation Message** from the drop-down list.
That's because we're creating a conversation message under the SR.
- **Action Type:** Select **Open a Dialog** from the drop-down list.

b. Request Details region:

The Request End Point is populated automatically, so you don't have to change anything here.

c. Response Details region:

You can leave this page blank. That's because clicking the **Reply to Service Request** button would show a message text box in Microsoft Teams. And you're not sending a response message from Service Center to Microsoft Teams.

d. Add Action Attribute region:

Here, you would add only one attribute, which is **Message Content**.

- **Attribute Name:** Select **MessageContent** from the drop-down list.
- **Data Type:** Select **String** from the drop-down list.
- **User Interface Component:** Select **Text Area** from the drop-down list.
- Select the **Included in Request** switch.

8. Click **Save**.

Note: Instead of clicking the **Reply to Service Request** button, the subject matter expert (SME) can also reply by clicking the native **Reply** action. The SME must mention the app in the format `@AppName` in the native Reply. Here, AppName is the name of your company's Microsoft Teams app created for Service Center. When the SME mentions your app in this format, your app adds their reply as a message on the request.

Create a Child Action for the Reply to Request Child Action

The child of the Reply to Service Request action in Microsoft Teams would be the Submit Reply action. When a Microsoft Teams user enters a message and clicks Submit Reply, the message goes to Redwood Service Center.

Note: You can use any other suitable action name instead of **Submit Reply**. Whatever action name you use is displayed in the Microsoft Teams interface.

To create the Submit Reply child action to the Reply to Service Request action:

1. Specify all the details for the child action as explained in Create a Collaboration Action. Considering the example of the **Reply to Service Request** button in the previous topic, you would specify the following details:

a. Action Details region:

- **Action Name:** Submit Reply
- **Action Code:** You can type any value of your choice here.
- **Business Object:** Select **Conversation Message** from the drop-down list.
- **Action Type:** Select **Call an API** from the drop-down list because it's calling a Service Center API.
- **Action Method:** Select **Post** from the drop-down list because you're creating a new message.
- **Display Sequence:** Select **1**.

b. Request Details region:

- **Request Content Type:** This field is populated automatically.
- **Request End Point:** This field is populated automatically.
- **Request Payload:** Add the following payload in this field:

```
{ "MessageContent": "[[MessageContent]]", "MessageTypeCd":  
"ORA_SVC_INBOUND", "ParentMessageId": [[ConversationMessageId]], "SenderAlternateId":  
[[ $system.CollabResourceId]], "SenderFreeFormAddress":  
"[[ $system.ExternalResourceDisplayName]]", "MessageAlternateId":  
"[[ $system.ExternalMessageId]]", "recipients": [[ $system.ConversationMessageRecipients]] }
```

c. Response Details region:

- **Response Visibility:** Select Public from the drop-down list.
- **Response Payload:** Set the value to: { "text": "[[MessageContent]]" }

This message content is also echoed to all the other SMEs in Microsoft Teams, so they all can view the reply given by the first SME. And they can add on to it if required.

- **Response Type:** Select **User-Defined JSON Formatted Result** from the drop-down list.
- **Response Content Type:** This field is populated automatically.

d. Add Action Attribute region:

This page has only one attribute, which is **Message Content** because that's the only attribute for Submit Reply.

- Ensure that you select the **Included in Request** switch.

2. Click **Save**.

Configure the FND_IDP_PROXY_USER_WHITELIST Profile Option

Let's say your agent sends a message to a Microsoft Teams user, seeking help to resolve a customer issue. When the Microsoft Teams user replies, their reply reaches Redwood Service Center only if the FND_IDP_PROXY_USER_WHITELIST profile option is set to the default value FUSION_APPS_CRM_SOA_APPID.

That's because the FUSION_APPS_CRM_SOA_APPID user's credentials are used to create all messages sent from Microsoft Teams to Redwood Service Center.

Note: Typically, the profile option is automatically set to the default value. But sometimes, it may have a blank value. If the value isn't set to FUSION_APPS_CRM_SOA_APPID, you can follow these steps to set the value.

(Optional) To configure the FND_IDP_PROXY_USER_WHITELIST profile option:

1. Sign in as a setup user or administrator.
2. Navigate to the **Setup and Maintenance** work area.
3. Open the **Tasks** panel tab and click **Search**.
4. Search for and select the **Manage Administrator Profile Values** task.
5. On the Manage Administrator Profile Values page, navigate to the Search region.
6. In the **Profile Option Code** field, type FND_IDP_PROXY_USER_WHITELIST.
7. Click **Search**.
The FND_IDP_PROXY_USER_WHITELIST profile option is displayed in the Search Results: Profile Options region.
8. In the FND_IDP_PROXY_USER_WHITELIST: Profile Values region, ensure that the **Profile Value** for the Site **Profile Level** is set to **FUSION_APPS_CRM_SOA_APPID**.
9. Click **Save**.

Upload Your Microsoft Teams App

To upload the Microsoft Teams app that you created for Redwood Service Center:

1. In the Microsoft Teams navigation pane, click **Apps**.
2. In the Apps page, click **Upload a custom app** from the list.
3. Click **Upload for app name**.
Here, app name stands for the name of the Microsoft Teams app that you created for Redwood Service Center.
4. In the **Open** dialog box, search for the compressed file for your Microsoft Teams app. You downloaded this compressed file in the last step in Create a Microsoft Teams App for Redwood Service Center.
5. Click **Open**.
The app that you created is now uploaded to Microsoft Teams.

Related Topics

- [Create a Microsoft Teams App](#)

How do I install the Microsoft Teams app and add it to my team?

After you upload your Microsoft Teams app, you must install the app and add it to your team in Microsoft Teams. Here's how:

1. In Microsoft Teams, go to the **Apps** tab and click **More apps**.

In the **Built by your org** region of the Apps page, you can see the Microsoft Teams app that you created for Service Center.

Note: If you've created multiple apps, you can see all them here.

2. Click the app that you want to install.
3. From the **Open** drop-down list, click **Add to a team**.

In the **Type a team or channel name** region, your channel is selected by default.

4. If you've created multiple apps, you can select the app of your choice by clicking the cross mark.
5. Click **Set up a bot**.

Your Microsoft Teams app is now installed for the team associated with the channel selected in step 3.

6. Navigate to **Teams** in the navigation bar.
7. Click the ellipsis next to your app name.
8. Click **Manage teams**.

A page with your team name is displayed.

9. Click the **Apps** tab.

Your Microsoft Teams app is now displayed in the list of apps.

26 Configure Notifications

Set Notification Triggers and Preferences

How do I enable notifications?

A notification is an alert for users such as service agents and managers, to notify them about an event on the service request, and enable them to take the necessary actions.

As a prerequisite, to allow users to view notifications, you must enable a profile option.

Enabling notifications is a global setup and feature opt-in step. Here's how you enable the notifications feature for service requests:

1. Sign in to the application as a setup or administrator user.
2. Navigate to **Setup and Maintenance**.
3. Select the **Service** offering.
4. In the **Functional Areas** section, select **Change Feature Opt In**.
5. Select the **Enable** check box for in the **Productivity Tools** row.
When the **Notifications** feature is enabled, bell notifications are automatically enabled for service requests.
6. Click the **Edit** icon in the **Features** column of the **Productivity Tools** row.
7. Enable your required notification delivery channels:
 - To enable push notifications for mobile devices, select the **Enable** check box for **Mobile Notifications**.
 - To enable browser notifications, select the **Enable** check box for **Browser Notifications**.
 - To enable email notifications, select the **Enable** check box for **Use Email as a Notification Delivery Channel**.
8. Click **Done** on the Edit Features Productivity Tools page.
9. Click **Done**.

Profile options to enable retries for the notification dispatcher

After removing a queue from an event, the notification dispatcher thread tries to enrich the event with all the details of the business object record.

If you've added extensive configurations and triggers, this process might occur before the record is created, causing a delay in record persistence. As a result, when the notification dispatcher tries to query the record before it has been persisted, an exception might be shown.

1. As an administrator, go to Setup and Maintenance, click the Tasks icon, search and select Manage Administrator Profile Values.
2. To set the number of retries, do the following:
 - a. Search ORA_SVC_NOTIFICATION_RETRY_COUNT in the Profile Option Code field.
 - b. In the Profile Values list, change the profile value to the number of retries you want. The default value is 3.

3. To set the delay time period (in seconds) for the retry, do the following:
 - a. Search `ORA_SVC_NOTIFICATION_RETRY_DELAY` in the Profile Option Code field.
 - b. In the Profile Values list, change the profile value to the number of seconds you want before the retry. The default value is 0.

Supported objects for sending service notifications using AI Notification Agent

The AI Notification Agent for Sales and Service can be triggered by other AI agents, such as the Escalation Agent, whenever you need to inform a service request owner by email or bell notification that their service request is being escalated.

The Notification Agent handles the details for deciding how to deliver the notification through the supported notification channels, offering an option for any other AI Agent to get a user's attention.

Supported objects are:

- Service Request
- HR Help Desk
- ISR
- Opportunity
- Lead
- Activity
- Account
- Contact

How do I set up groovy notification triggers?

As an administrator, here's how you configure notifications after you've enabled the notifications feature:

1. 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, a trigger can be defined to send a notification when a high severity SR has been assigned to an agent.

For more information about defining notification triggers, see the "Define Notification Triggers" section.

2. Configure the notification preferences using the Notification Preferences page. For example, to specify the notification message using SmartText, and the notification recipient.

For more information about notification preferences, see the "Set Notification Preferences" section.

Note:

- You can temporarily disable all notifications using the profile option SVC_DISABLE_BO_NOTIFICATIONS. Oracle recommends to use this option when you're doing a bulk import of data, so that you don't get multiple notifications.
- You can temporarily enable or disable specific notifications from the Notification Preferences page.

Understand the Two sendNotification Groovy Functions

You use this first example function when there is no Notification Preferences support for overriding messages, recipients or channels.

```
adf.util.sendNotification(java.lang.Object, java.util.HashMap\
```

Sends a notification to all enabled delivery methods. The delivery methods, message text, and recipients can be configured on the notification preferences page. If notification preferences for the trigger exist, then the values in the preferences take precedence. The first argument is the ADF object. The second argument is the requestParameters HashMap, which can specify one or more of the following parameters: 'Channels', 'MessageText', 'RecipientPartyId', 'RecipientPartyIdList', 'RecipientAttributeNames'.

Example Invocation:

```
def map = new HashMap();  
map.put("Channels", ["ORA_SVC_BELL", "ORA_SVC_MOBILE"]);  
map.put("MessageText", "Map defaulted message"); map.put("RecipientPartyId",  
AssigneeResourceId); adf.util.sendNotification(adf, map); Example Invocation  
2: def map = new HashMap(); map.put("Channels", ["ORA_SVC_BELL"]);  
map.put("RecipientPartyIdList", [AssigneeResourceId, ReportedByPartyId]);  
adf.util.sendNotification(adf, map); Example Invocation 3: def map = new  
HashMap(); map.put("Channels", ["ORA_SVC_BELL"]);  
map.put("RecipientAttributeNames", ["AssigneeResourceId",  
"ReportedByPartyId"]); adf.util.sendNotification(adf, map);
```

The second example function is used when there's Notification Preferences support to override messages, recipients, and channels.

```
adf.util.sendNotification(java.lang.Object, java.lang.String, java.lang.String, java.lang.String, java.lang.Long)
```

Sends a push notification to all enabled channels, such as mobile or web browser bell. The channels, default messages, recipients, and other preferences can be configured on the notification rules page. A notification preference is associated with the objectCode and eventName pair. If notification preferences for the trigger exist, then the values of messageText and partyId in the preferences take precedence. The first argument is the adf object. The second argument is the eventName, which should typically match the trigger name. The third argument is the objectCode, which should typically match the business object name, for example, ServiceRequestVO. The fourth argument is the messageText. The fifth argument is the partyId.

Example Invocation:

```
adf.util.sendNotification(adf, 'SRCritical', 'ServiceRequestVO', 'Service Request is now critical!', partyId).
```

Note: If you use this function and do not set up a Notification Preference, it will by default send the notification to both the bell and push channels.

Custom Objects or Fields

1. In Application Composer, navigate to **Service Request > Fields > Create a Custom Field**.
2. Navigate to **Pages** and include the new field in **Creation Page Layout**, and **Details Page Layout**.
3. Write your trigger.
Check the API Name of your new field. If you made a field called Priority, the API name will be something like **Priority_c**.

How do I define notification triggers?

Here's how you define a Groovy script for a notification trigger:

1. Sign in to the application as a setup or administrator user.
2. Click **Navigator > Configuration > Sandboxes**.
3. Create a sandbox or select an existing one, and click **Enter Sandbox** to activate the sandbox. Your sandbox is activated, and you can see its name on the sandbox bar over the global header.
4. In the **Navigator**, click **Application Composer**. The Application Composer page is displayed.
5. Select **Objects > 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.
 - Oracle recommends you to use the trigger type **After Changes Posted to Database**. This trigger type enables you to stop potential issues if the Groovy script is accidentally written to run indefinitely. If the trigger type is set to **Before Update to Database**, with a bad script, there may be some issues.
 - If you're creating a new object, and you want to trigger a notification when the object is created, Oracle recommends you to use the trigger type **Before Insert to Database**. However, some of the **Before** trigger types don't have all attributes exposed yet, resulting in some fields being blank. To debug your triggers if you're not getting the expected results, follow the steps in the section "Debug Groovy Triggers".
 - 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 update, and then comparing the two values to see whether the attribute is changed. However, this workaround works only for high-level attributes such as `criticalFlag` and `status`. This workaround may not work for the `viewRow` attribute, for example, when you're retrieving the team from an SR.

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:

- i. In the **Trigger Definition** section, click **Show/Hide Expression Palette**.
- ii. Click the **Fields** tab.
- iii. Select an **Object**.
- iv. Click the **Maximize Edit Script** arrow. The fields for the selected object are displayed.

- v. Select the API you want, and click **Insert**.
 - vi. To close the expression palette, click the **Restore Edit Script** arrow, and then click **Show/Hide Expression Palette**.
- b. Click **Save and Close**.
- 9. Navigate to the Notification Preferences page to configure your preferences for the notifications.

For more information about configuring notification preferences, see the "Set Notification Preferences" topic.

Note: You can reuse a Groovy notification trigger if no other notification uses it.

How do I set notification preferences?

The Notification Preferences page enables administrators to configure the notification rules. Users who have the ORA_SVC_SR_ADMINISTRATOR or ORA_SVC_SR_POWER_USER duty roles can access this page.

Notification preferences are used to select multiple recipients like team members, queue members, and so on. When notification preferences is used, it will override the recipients that are set in the Groovy code.

Here's how you configure the notification rules:

Note: Ensure that your triggers are published.

1. Sign in to the application as an administrator.
2. In the **Navigator**, click **Tools > Notification Preferences**.

The Notification Preferences page is displayed.

3. From the **Object** drop-down list, select **Service Request** or **Work Order**.

Note: Other supported objects are also displayed in the **Object** drop-down list.

- a. Click **Add**. A blank row is added in the table.
- b. To enable the notification trigger, select **Yes** from **Enabled** drop-down list.
- c. To prevent users from personalizing this notification through the User Notification Preferences page, select **No** from the **OverrideFlag** drop-down list.

Note: To clean up the user notification preferences data, in case the resource leaves your company or is no longer active, run the **Purge Obsolete User Preferences** scheduled process. This scheduled process lets your organization ensure that your data on user notification preferences isn't stale.

- d. Select a Groovy notification trigger from the **Triggering Event** drop-down list.
- e. Enter the **Notification Name** and **Description**.
- f. In the **Recipients** column, click **Edit**.

The **Configuration for Trigger Name** dialog box is displayed.

- g. Select the notification delivery options for the respective recipients.

For an enabled triggering event, notifications are sent to the specified recipients only if you select at least one delivery option. If any delivery method is selected for a specific recipient, that recipient will also receive bell notifications. Also, if you don't select the **Bell Notifications** option, then no notifications will be sent to the specified recipients.

The following tables describe the recipients and delivery options for the service request object:

Recipients for Service Request Object	Delivery Options for Service Request Object
- Assigned To: Resource to whom the service request is assigned. - Manager of Assigned To: Manager of the resource to whom the service request is assigned. If a service level agreement is about to expire, and immediate attention is required, the manager of the resource to whom the service request is assigned, can be informed. - Resource Team: Extra resources that have been added to a service request on the Team subtab. If you've a team of resources collaborating on a service request, and a trigger such as the service request escalating to Critical status happens, the entire team can be informed. - Queue Owner: Owner of the queue. If your organization uses manual assignment, the queue owner can be notified when a new service request is associated with a queue owned by the queue owner. For information about the setup steps required to send a notification to a queue owner, see the "Trigger a Notification to the Queue Owner" section. - All Queue Members: Send defined notifications to all members of a specified queue, so that they can effectively monitor the queue for open items. Queue notifications reduce the need for agents to manually check each queue they're associated with, to see whether there's new work. To enable this recipient option, set the profile option <code>ORA_ENABLE_QUEUE_MEMBER_NOTIFICATIONS</code> to Yes. This	- Bell Notifications: Sends a bell notification to the web application. - Mobile Notifications: Sends a mobile push notification to the Oracle Fusion Cloud Mobile application. - Browser Notifications: Sends a real-time notification to the desktop of the selected users that are signed in the Omnichannel toolbar, when a predefined trigger is executed. - Email Notifications: Sends an email notification to the selected internal recipients' primary work email ID saved in the application. Note: The from email address is configured through the <code>SVC_OUTBOUND_EMAIL_FROM</code> profile option. If that isn't specified, the default from email address <code>noreply@oracle.com</code> is used.

Recipients for Service Request Object	Delivery Options for Service Request Object
profile option is disabled by default. You should only enable this option if your queues have a few participants. If your organization's queues have many members, enabling notifications to all of them might affect application performance.	

The following tables describe the recipients and delivery options for the Work Order object:

Recipients for Work Order Object	Delivery Options for Work Order Object
- Assigned To: Resource to whom the work order is assigned. - Created By: The user who created the Work Order - Field Service Resource ID: The resource ID being used to complete the work order - Last Updated By: The user who last updated the work order. - Manager of Assigned To ID: Manager of the resource to whom the work order is assigned. If a service level agreement is about to expire, and immediate attention is required, the manager of the resource to whom the work order is assigned, can be informed. - Requested By: The user who reported this work order.	- Bell Notifications: Sends a bell notification to the web application. - Browser Notifications: Sends a real-time notification to the desktop of the selected users that are signed in the Omnichannel toolbar, when a predefined trigger is executed. - Email Notifications: Sends an email notification to the selected internal recipients' primary work email ID saved in the application. Note: The from email address is configured through the SVC_OUTBOUND_EMAIL_FROM profile option. If that isn't specified, the default from email address <code>noreply@oracle.com</code> is used.

- h. To let users follow a specific service request and receive all notifications for that service request, select the **Enable Followers** check box.

When enabled, users with the correct permissions will see the **Follow** option in the **Actions** menu of the Service Request Details page.

The notification followers receive notifications for all supported channels, when any notification event is triggered for the service requests they follow.

To enable this feature for the site, set the `ORA_ENABLE_FOLLOW_NOTIFICATIONS` profile option to **Yes**.

To follow a service request, users must have the `SVC_GET_SR_FOLLOW_NOTIFICATIONS` permission.

This privilege is added to the following ready-to-use duty roles:

- Service Request Administrator
- Service Request Channel User
- Service Request Contributor
- Service Request Power User
- Service Request Troubleshooter

Note: To evaluate whether a resource is still valid and enabled to follow specific service requests, run the **Purge Obsolete Notification Followers** scheduled process. This scheduled process removes inactive and end-dated users from following the service request, so that the application doesn't get overloaded with stale data. This scheduled process lets your organization ensure that your data on notification followers isn't stale.

- i. Click the **New SmartText** link and enter the **Notification Text** for the selected object.

For more information about using SmartText, see the "Using SmartText" topic.

4. (Optional) To delete a notification preference, select the row and click **Delete**. The associated notification text is also deleted.

Note: If you delete a notification that uses a Groovy notification trigger, you can create a new notification using the same trigger, if no other notification uses it.

5. (Optional) To change an existing notification text, click the **Update SmartText** icon for the selected row.
6. Click **Save**.

Define Notifications for Custom Objects

The Notification Preferences page lets you define notifications not only for standard objects, but for custom objects too. First, use Application Composer to create triggers on the relevant custom objects for which you want to initiate user notifications. Then, use the Notification Preferences page to specify the recipients and delivery channels for these custom objects.

Here's how you set up notifications for custom objects:

1. Sign in to the application as a setup or administrator user.
2. Click **Navigators > Configuration > Sandboxes**.
3. Create a sandbox or select an existing one, and click **Enter Sandbox** to activate the sandbox. Your sandbox is activated, and you can see its name on the sandbox bar over the global header.
4. Publish the selected sandbox.

5. In the **Navigator**, click **Application Composer**. The Application Composer page is displayed.
6. Select **Objects > Custom Objects**.
7. Select an existing custom object or create a new custom object with default fields and pages.

For more information on working with custom objects and tags, see *Add Objects and Fields* in the Configuring Applications Using Application Composer guide.
8. On the Object Overview page, click **Manage Tags**.

The **Object Tags** window is displayed.
9. Select the **Notifications** check box and click **Ok**.
10. Navigate to **Server Scripts** for the selected custom object.

The Server Scripts page for the selected custom object is displayed.
11. Click the **Triggers** tab, and create a new trigger.
12. Click **Navigator > Tools > Notification Preferences**.

The Notification Preferences page is displayed, where you can configure your notification preferences for the selected custom object.
 - a. From the **Object** drop-down list, select the custom object.
 - b. Select the custom object triggering event, and specify the notification recipients and delivery options.
 - c. Click **Save**.

Trigger a Notification to the Queue Owner

Here's how you trigger a notification to the queue owner:

1. Add the **Owner** field to the Queue pages as described in the "Add the Queue Owner to the Page Layouts for Queues" topic.
2. Define the notification preference for the **Service Request** object:
 - a. Click **Standard Objects > Service Request > Server Scripts**.

The Server Scripts Service Request page is displayed.
 - b. Create an object trigger for service request escalated. For a sample Groovy script, see the "Sample Groovy Scripts for Notifications" section.
 - c. In the **Navigator**, click **Tools > Notification Preferences**.

The Notification Preferences page is displayed.
 - d. Create a notification preference using the service request escalated trigger. For the **Recipients** option, select only **Queue Owner**.
3. Create a queue for associating the queue owner with a resource:
 - a. On the Home page, click **Service > Queues**.
 - b. On the Queues page, create a queue and specify a **Name** for the queue.
 - c. From the **Owner** drop-down list, select a resource member.
 - d. Click **Save and Close**.
4. On the Create or Edit Service Request page, from the **Queue** drop-down list, select the queue name you specified in the previous step.
5. Trigger a notification to the specified queue owner by enabling or disabling the **Critical** status.

Related Topics

- [Overview of Sandboxes](#)
- [Create and Activate Sandboxes](#)
- [Publish Sandboxes](#)
- [How do I add the queue owner field to the page layout?](#)

Debug Groovy Triggers

Within your Groovy scripts, all your `println` statements go to the runtime messages. Here's how you view the runtime log and debug your Groovy script:

1. Within a sandbox, navigate to **Application Composer**.
2. Select **Common Setup > 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.

Related Topics

- [Overview of Sandboxes](#)
- [Create and Activate Sandboxes](#)
- [Publish Sandboxes](#)

Sample Groovy Scripts for Notifications

How do I use Groovy script to notify an agent that a service request has been assigned?

Here's a sample groovy script that triggers a notification to be sent to the recipient specified in the Notification Preferences page when a service request (SR) is assigned.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Service Request > Server Scripts.

```
/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Update in Database
 * OBJECT: Service Request
 * Use Case: Send a notification when an existing SR is reassigned.
 * Note: You can override the Notification Text and Recipients using the Notification Preferences page.
 */

if (isAttributeChanged('AssigneeResourceId')) {
  try {
    def recipientPartyId = AssigneeResourceId
    def messageText = 'An SR notification (default message).'
```



```
if (recipientPartyId) {
    //Call to send notification
    adf.util.sendNotification(adf, messageText, recipientPartyId)

    //Log a confirmation that the notification has been sent. Logs can be viewed in 'Runtime Messages'.
    //println("Notification sent to " + recipientPartyId + " because the SR was assigned to them.")
} else {
    println("No Assignee associated with this SR")
}
} catch (e) {
    //Log the failure in groovy logging. Logs can be viewed in 'Runtime Messages'.
    println("Failure to trigger notification from Groovy Script " + e.getMessage());
}
// Throwing validation exception will show the message on the UI. This is not recommended for published
sandboxes.
// The following code is one of many to illustrate identifying an error in trigger from the UI.
// Replace <triggerName> with the trigger name you specified when creating this trigger.
// throw new oracle.jbo.ValidationException('Failure to trigger <triggerName> Notification from Groovy
Script: ' + e.getMessage())
}
}
```

Service Request Escalated

Here's a sample code to trigger a notification to be sent to the recipient specified in the Notification Preferences page when the SR is escalated.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Service Request > Server Scripts.

```
/* DISCLAIMER: This trigger is provided only as a reference.
* TRIGGER TYPE: Before Update in Database
* OBJECT: Service Request
* Use Case: Send a notification when an SR is escalated.
* Note: You can override the Notification Text and Recipients using the Notification Preferences page.
*/

if (isAttributeChanged('CriticalFlag') && CriticalFlag=='Y') {
    try {
        def recipientPartyId = AssigneeResourceId
        def messageText = 'An SR notification (default message).'

        if (recipientPartyId) {
            //Call to send notification
            adf.util.sendNotification(adf, messageText, recipientPartyId)

            //Log a confirmation that the notification has been sent. Logs can be viewed in 'Runtime Messages'.
            //println("Notification sent to " + recipientPartyId + " because the SR was escalated.")
        } else {
            println("No Assignee associated with this SR")
        }
    } catch (e) {
        //Log the failure in groovy logging. Logs can be viewed in 'Runtime Messages'.
        println("Failure to trigger notification from Groovy Script " + e.getMessage());
    }

    // Throwing validation exception will show the message on the UI. This is not recommended for published
    sandboxes.
    // The following code is one of many to illustrate identifying an error in trigger from the UI.
    // Replace <triggerName> with the trigger name you specified when creating this trigger.
    // throw new oracle.jbo.ValidationException('Failure to trigger <triggerName> Notification from Groovy
    Script: ' + e.getMessage())
}
```

}

How can I be notified when a customer replies to a message?

Here's a sample code to trigger a notification to be sent to the recipient specified in the Notification Preferences page when a customer replies to a message.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Service Request > Message > Server Scripts.

```
/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Insert to Database
 * OBJECT: Message (child of Service Request)
 * Use Case: Send a notification to the assignee when a customer entry is created.
 * Note: You can override the Notification Text and Recipients using the Notification Preferences page.
 */
if (MessageTypeCd == 'ORA_SVC_CUSTOMER_ENTRY') {
    try {

        // Note: The notation ServiceRequest?.SrNumber is used because ServiceRequest is the parent
        // object of Messages. To access the other possible field names, expand the trigger box, click on fx,
        // click on Fields, and select the drop-down list from Message. Click on Service Request to see the
        // available fields. You must click Insert to correctly insert the appropriate field into your script.

        def messageText = "A customer entry was received for " + ServiceRequest?.SrNumber
        def recipientPartyId = ServiceRequest?.AssigneeResourceId
        if (recipientPartyId) {
            // Call to send notification
            adf.util.sendNotification(adf, messageText, recipientPartyId)
            //Log a confirmation that the notification has been sent. Logs can be viewed in 'Runtime Messages'.
            //println("Notification sent to " + recipientPartyId + " because a customer entry was received.")
        } else {
            println("No Assignee associated with this SR")
        }
    } catch (e) {

        // Log the failure in groovy logging. Logs can be viewed in 'Runtime Messages'.
        println("Failure to trigger notification from Groovy Script " + e.getMessage());

        // Throwing validation exception will show the message on the UI. This is not recommended for published
        // sandboxes.
        // The following code is one of many to illustrate identifying an error in trigger from UI.
        // Replace <triggerName> with the trigger name you specified when creating this trigger.
        // throw new oracle.jbo.ValidationException('Failure to trigger <triggerName> Notification from Groovy
        // Script: ' + e.getMessage())
    }
}
```

Can I view an expired milestone sample script?

This is sample code to trigger a notification to be sent to the recipient specified in the Notification Preferences page when a milestone expires.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Service Request > Milestone > Server Scripts.

Note: This example uses `WarnedFlag`. However, you can also use `ComplianceFlag` for similar results.

```
/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Update in Database
 * OBJECT: Milestone (child of Service Request)
 * Use Case: Send a notification to the assignee when an SR milestone is expiring.
 * Note: You can override the Notification Text and Recipients using the Notification Preferences page.
 */

if (isAttributeChanged('WarnedFlag') && WarnedFlag=='Y') {
  try {
    def recipientPartyId = SrAssigneePartyId
    def messageText = 'An SR milestone notification (default message).'

    if (recipientPartyId) {
      //Call to send notification
      adf.util.sendNotification(adf, messageText, recipientPartyId)
      //Log a confirmation that the notification has been sent. Logs can be viewed in 'Runtime Messages'.
      //println("Notification sent to " + recipientPartyId + " because the SR was escalated.")
    } else {
      println("No Assignee associated with this SR")
    }
  } catch (e) {
    //Log the failure in groovy logging. Logs can be viewed in 'Runtime Messages'.
    println("Failure to trigger notification from Groovy Script " + e.getMessage());
  }

  // Throwing validation exception will show the message on the UI. This is not recommended for published
  // sandboxes.
  // The following code is one of many to illustrate identifying an error in trigger from the UI.
  // Replace <triggerName> with the trigger name you specified when creating this trigger.
  // throw new oracle.jbo.ValidationException('Failure to trigger <triggerName> Notification from Groovy
  // Script: ' + e.getMessage())
}
}
```

Work Order Assigned

This is sample code to trigger a notification to be sent to the recipient specified in the Notification Preferences page when a work order is created.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Work Order > Server Scripts.

```
/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Insert to Database
 * OBJECT: Work Order
 * Use Case: Send a notification when a work order is created.
 * Note: You can override the Notification Text and Recipients using the Notification Preferences page.
 */

try {
  def messageText = WoId.toString() + ':' + WoNumber + ' - This Work Order has been assigned to you'
  def recipientPartyId = AssigneeResourceId
  def recipientName = AssigneePersonName
  if (recipientPartyId) {
    adf.util.sendNotification(adf, messageText, recipientPartyId)
    //Log a confirmation that the notification has been sent. Logs can be viewed in 'Runtime Messages'.
    //println("Notification sent to " + recipientName + " because Work Order assigned to them")
  } else {
    println("No Assignee associated with this Work Order")
  }
}
```

```

    }
    } catch (e) {
        //Log the failure in groovy logging. Logs can be viewed in 'Runtime Messages'.
        println("Failure to trigger notification from Groovy Script " + e.getMessage());

        // Throwing validation exception will show the message on the UI. This is not recommended for published
        // sandboxes.
        // The following code is one of many to illustrate identifying an error in trigger from the UI.
        // Replace <triggerName> with the trigger name you specified when creating this trigger.
        // throw new oracle.jbo.ValidationException('Failure to trigger <triggerName> Notification from Groovy
        // Script: ' + e.getMessage())
    }
}

```

Work Order Complete

This sample code triggers a notification to be sent to the recipient specified in the Notification Preferences page when a work order status is updated to complete.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Work Order > Server Scripts.

```

/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Update to Database
 * OBJECT: Work Order
 * Use Case: Send a notification when a work order is completed.
 * Note: You can override the Notification Text and Recipients using the Notification Preferences page.
 * When WO_STATUS_CD is exposed in Application Composer, this script can be changed. Change everywhere that
 * says 'WoStatusCdMeaning' to 'WoStatusCd'
 * and change the comparison in the if statement to 'ORA_SVC_WO_COMPLETE' instead of 'Complete'.
 */

if (isAttributeChanged('WoStatusCdMeaning') && WoStatusCdMeaning == 'Complete') {
    try {
        def messageText = WoId.toString() + ':' + WoNumber + ' - This Work Order has been Completed'
        def recipientPartyId = AssigneeResourceId
        def recipientName = AssigneePersonName
        if (recipientPartyId) {
            adf.util.sendNotification(adf, messageText, recipientPartyId)
            //Log a confirmation that the notification has been sent. Logs can be viewed in 'Runtime Messages'.
            //println("Notification sent to " + recipientName + " because status of WO is Complete")
        } else {
            println("No Assignee associated with this Work Order")
        }
    } catch (e) {
        //Log the failure in groovy logging. Logs can be viewed in 'Runtime Messages'.
        println("Failure to trigger notification from Groovy Script " + e.getMessage());

        // Throwing validation exception will show the message on the UI. This is not recommended for published
        // sandboxes.
        // The following code is one of many to illustrate identifying an error in trigger from the UI.
        // Replace <triggerName> with the trigger name you specified when creating this trigger
        // throw new oracle.jbo.ValidationException('Failure to trigger <triggerName> Notification from Groovy
        // Script: ' + e.getMessage())
    }
}

```

Send Internal Note

This is sample code to trigger a notification to be sent to a recipient when an new SR message of type Internal Note is created.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Service Request > Message > Server Scripts.

```
/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Insert to Database
 * OBJECT: Message (child of Service Request)
 * Use Case: Send an internal notification to the user when an SR message is created.
 */

if(MessageTypeCd == 'ORA_SVC_INTERNAL_NOTE')
{
    try{
        def map = new HashMap();
        map.put("Channels", ["ORA_SVC_BELL", "ORA_SVC_EMAIL"]);
        map.put("MessageText", "Internal Note message");
        map.put("RecipientPartyId", ServiceRequest?.AssigneeResourceId);
        map.put("ObjectCode", 'ServiceRequestVO');
        map.put("ObjectId", SrId);
        adf.util.sendNotification(adf, map);
    }
    catch (e) {
        throw new oracle.jbo.ValidationException('Failure: ' +
e.getMessage())
    }
}
```

Add Service Request Resource

This is sample code to trigger a notification to a new resource member when that member is added to an SR.

You can copy and paste this sample code in Application Composer > Objects > Standard Objects > Service Request > Service Request Resource > Server Scripts.

```
/* DISCLAIMER: This trigger is provided only as a reference.
 * TRIGGER TYPE: Before Insert to Database
 * OBJECT: Service Request Resource (child of Service Request)
 * Use Case: Send a notification to a new resource member when that member is added to an SR.
 */

try {
    def map = new HashMap();

    // Specify default MessageText
    def messageText = "You are now a team member"

    // Specifying the recipient
    def recipientPartyId = PartyPartyId

    //Resolving Parent child object relation. Use respective parent object apinames. Eg:
    ServiceRequestVO,HRHelpDeskRequestVO
    def objectCode='ServiceRequestVO'

    Long objectId = SrId
```

```
// Specify one or more channels
map.put("Channels", ["ORA_SVC_BELL", "ORA_SVC_EMAIL"]);

map.put("MessageText", messageText);
// The following can be used to pass a Long PartyId
map.put("RecipientPartyId", recipientPartyId);
map.put("ObjectCode", objectCode);
map.put("ObjectId", objectId)
if ( recipientPartyId != null ) {
// Call to send notification
adf.util.sendNotification(adf, map);
}
else {
println("No recipient. Notification will not be sent.");
}
}
catch (e) {
// Log the failure in groovy lo'RuntimeMessages'.
println("Failure to trigger notification from Groovy Script " +
e.getMessage());
}
```

Set Push Notifications

Configure Push Notifications

With the push notification framework users, such as service agents and managers, can get push notifications about service request events delivered to their mobile devices. Once an agent is signed in to the mobile application, they can view a list of their notifications by

You define notifications in Application Composer with your unique conditional logic. This lets you decide when to raise each notification. For example, you can define alerts for upcoming SLA milestones, reassignments, or escalations. The administrator can also decide whether to enable mobile push notifications for certain alerts, or to deliver the alerts only to the Bell notifications on the desktop application. Note that bell notifications are always delivered, even if mobile notifications are disabled. Clicking a notification and navigating to the record details marks the notification as read, and it's cleared from the notification list.

You define notification triggers as a groovy expression in Application Composer using conditions that must be met for each notification. For example, a trigger can be defined to send a notification when a high-severity service request is assigned to a user. When the Service Notifications feature is enabled, Bell notifications are automatically enabled.

Push notifications are supported in iOS and Android devices.

Here's what you do to enable push notifications:

1. Log in as an administrator, click Setup and Maintenance, and then from the Setup menu, select **Service**.
2. In the Functional Areas section, select Productivity Tools, and then select **Change Feature Selection**.
3. Select Service Notifications, then click the **Enable** check box to enable Service Notifications and Mobile Notifications.

27 Adaptive Intelligence

Overview of Adaptive Intelligence

Adaptive Intelligence enables you to augment your service team capabilities with intelligent, AI-powered features that are embedded directly within Fusion Service Center.

You can adopt them into your everyday workflows and boost your customer experience by delivering a more nimble and efficient response to service requests.

Use attachment attributes in Adaptive Search of service requests

You can use attachment field attributes such as file name and creation date in Adaptive Search to make searching for service requests easier.

This is helpful if, for instance, an agent remembers only attributes of an SR rather than the specific SR number. Enabling SRs to be discovered based on the attributes of their attachments can aid in operational efficiency and agent productivity. Here's how you enable it:

1. Sign in with the Application Implementation Consultant role.
2. From Setup and Maintenance, click the **Tasks** icon, and select **Search**.
3. Search and select **Configure Adaptive Search**.
4. Click the **Setup** tab, then select the **Advanced** subtab.
5. Expand the **Service Request** object, then expand **Attachments** and select **Attachment**.
6. Select the fields you want to make available for Adaptive Search by selecting the **Include in Keyword Search** checkbox.
7. From the Actions drop-down list select **Full Publish**.
8. Click **Save and Close**.

Configure Service Request Similarity

Enable Service Request Similarity

You must enable the SR Similarity feature before you can start using it in the application.

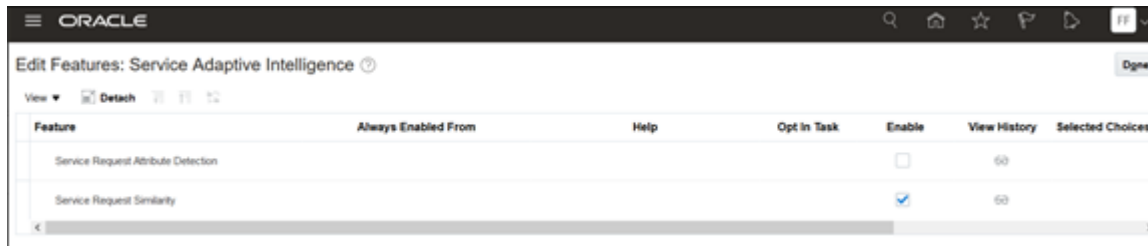
Here's how you do it.

1. Click **Navigator > My Enterprise > Offerings**.

2. On the Offerings page, select the **Service** offering.
3. Click the **Opt In Features** button.
4. On the Opt In: Service page, select **Enable** checkbox for the **Service Adaptive Intelligence** feature.
5. Click the **Features** (pencil) icon.

The Edit Features: Service Adaptive Intelligence page is displayed.

6. Select the **Enable** checkbox for the **Service Adaptive Intelligence** feature.

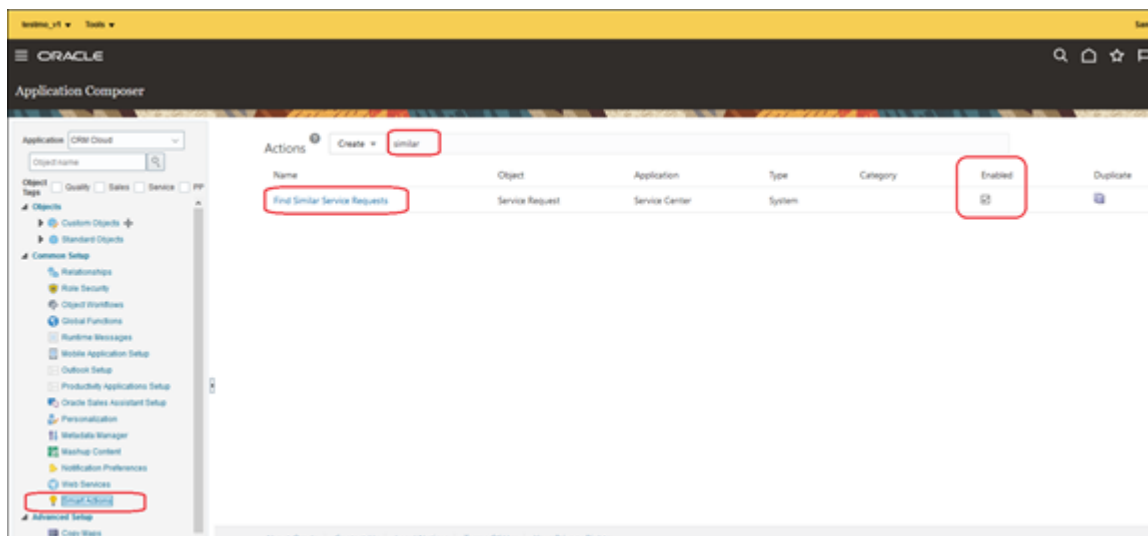


7. Click **Done**, and then on the Opt In: Service page, click **Done** again.
8. Now, switch the back end to point to the Spectra microservice by setting the profile option: SVC_SIMILAR_SR_REST_ENDPOINT to `https://<fusion_host_name>`. For example, `https://fa-xatt-pintlbfaddev.fa.ocs.oc-test.com`. Spectra service must be enabled in all environments where you want to enable the feature.

The Service Request Similarity feature is now activated.

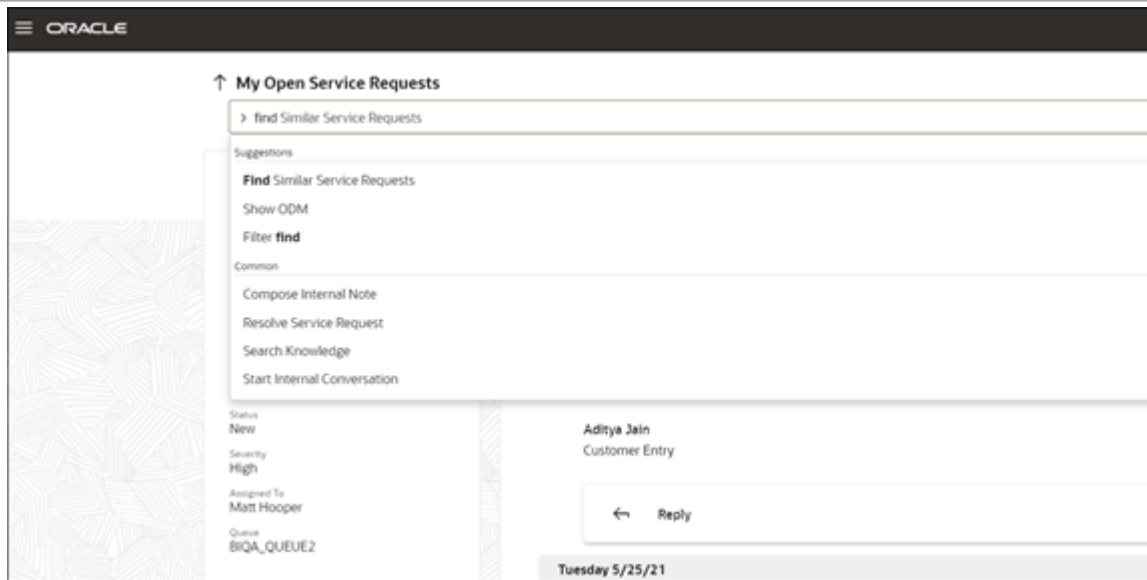
Note: You can also enable this feature by setting the SVC_ENABLE_SIMILAR_SR profile option to **Yes**. If you're Fusion Service Classic, you're done. If setting up Service Center, then continue with the following steps:

9. Click **Navigator > Configuration > Sandboxes** and create a sandbox.
10. Open Application Composer and click **Smart Actions**.
11. Search for and enable **Find Similar Service Requests** by selecting the **Enabled** checkbox.



12. Publish the sandbox.

The Find Similar Service Requests action in the Service Center Action bar will be available on the Service Request Overview page, in the Action Bar.



Configure Service Request Classification

What's Service Request Similarity?

Service Request (SR) Similarity enables an agent to troubleshoot issues accurately and efficiently, by finding similar SRs that may have been previously resolved.

SR Similarity uses adaptive intelligence (AI) for similarity computation, by considering the key standard or custom fields of an SR. The feature is seamlessly integrated within the SR review and edit workflow on the agent-facing user interface (UI).

The feature is currently applicable for English language implementations only.

Before deploying the feature, administrators and power users can test and evaluate similarity computation using the Administrator UI provided specifically for this purpose.

Prerequisites for Classifying Requests

Here's a list of prerequisites required for classifying service requests (SRs) created from inbound emails:

- Assign the required roles to your administrators as described in *Required Roles*.
- Ensure that you've enough data for resolved requests in your application. You must have a minimum of 50 records per category to start training the model and at least 2 categories. For the best results, Oracle recommends that you train the model with at least 10,000 records to have quality predictions. Each record must have a distinct category or product as required.

Note: The quality of predictions depends on the volume and quality of data that's used to train the model.

Required Roles

To activate service request (SR) classification in the Classification administration UI, you must first assign the following roles to your administrators:

- **CRM SVC Attribute Detection Administrator:** You can find this role in the application for AI Apps in Oracle Identity Cloud Service (IDCS).
- **Adaptive Intelligence Applications Administrator:** You can add this role from the Security Console.

Users who are assigned these roles can access the REST service of SR Classification Prediction Service and the administration UI of SR Classification.

Required Connections and Validations

The following pages help you to make the required connections and validations, so that Fusion Service can detect request attributes and make predictions.

Pages for the Required Connections and Validations

Page	What it does
Configure SR Classification	Enables the AI application to connect to Fusion Service. You must specify your Fusion Service user name and password here. These details are used to pull in the data from Fusion Service. For more information about this page, see Start the Data Ingestion and Model Training.
Inbound Email Configuration and Validation page in Fusion Service	Validates whether the configuration is complete, the model is built, and the prediction service works effectively. After the applications are connected, you can create a sample payload and see a sample prediction. You can test the AI feature to validate whether the Classification API prediction is correct. For more information about this page, see Validate the Setup.

Set Up Fusion Service for Service Request Classification

Steps to Set Up Request Classification

Here's the sequence of procedures that you must complete to set up classification for service requests:

Note: You must complete each step in the following order:

Setup Steps

Step	Topic
1.	<i>Enable Request Classification</i>
2.	<i>Configure Profile Options</i>
3.	<i>Validate Authentication with the AI Application</i>
4.	<i>Start the Data Ingestion and Model Training</i>
5.	<i>Validate the Setup</i>

Enable Request Classification

You must enable the classification of service requests created from inbound emails.

1. Sign in to Fusion Service as a setup user.
2. Navigate to the **Setup and Maintenance** work area and select the **Service** offering.
3. Click **Change Feature Opt In**.
4. On the Opt In: Service page, click the **Service Adaptive Intelligence** row and select the **Enable** check box.
5. On the same row, click **Edit** in the **Features** column.
6. On the Edit Features: Service Adaptive Intelligence page, click the **Enable** check box for **Service Request Classification**.
7. Click the **Email Classification** row and select the **Enable** check box.
8. Click **Done**.
9. Click **Done** on the Opt In: Service page.

Configure Profile Options

You can set options for classifying service requests (SRs) by configuring some email profile options. These profile options are predefined, but you can change their values based on your requirements.

Here are the profile options to configure followed by the procedure for configuration:

Profile Options for Classifying Service Requests

Profile Option	Description
ORA_SVC_ENABLE_ATTR_DETECTION_MASTER_IDCS_APP	Enables the use of the master IDCS application authentication for SR classification REST API calls. The default value is No. CAUTION: You must enable this profile option by setting the value to Yes, otherwise Fusion Service can't connect to the AI application.
SVC_ENABLE_INBOUND_MSG_SR_CATEGORY_PREDICTION	Enables the prediction for the category attribute during the processing of inbound emails to create SRs. The default value is Yes. To turn off category predictions, set the value to No.

Profile Option	Description
SVC_ENABLE_INBOUND_MSG_SR_PRODUCT_PREDICTION	Enables the prediction for the product attribute during the processing of inbound emails to create SRs. The default value is Yes. To turn off product predictions, set the value to No.
SVC_INBOUND_MSG_PREDICTION_THRESHOLD_FOR_SR_CATEGORY_ID	Specifies the threshold value to decide whether the predicted category ID can be stamped on the SR. The default value is: 0.6 This means that if the prediction has a confidence level of more than 60%, only then you stamp the predicted value on the SR. Otherwise, don't stamp it on the SR. It's possible to run the service in the dark mode by setting the threshold value to 1.0. In the dark mode, the prediction happens but it isn't stamped on the SR. This way, you can see what the correct labels would have been if the accuracy had been set lower.
SVC_INBOUND_MSG_PREDICTION_THRESHOLD_FOR_SR_PRODUCT_ID	Specifies the threshold value to decide whether the predicted product ID can be stamped on the SR. The default value is: 0.7 This means that if the prediction is more than 70% accurate, only then you stamp the predicted value on the SR. Otherwise, don't stamp it on the SR. It's possible to run the service in the dark mode by setting the threshold value to 1.0. In the dark mode, the prediction happens but it isn't stamped on the SR. This way, you can see what the correct labels would have been if the accuracy had been set lower.

1. Sign in as a setup user or administrator.
2. Navigate to the Setup and Maintenance work area.
3. Open the Tasks panel tab and click **Search**.
4. Search for and click the **Manage Administrator Profile Values** task.
5. On the Manage Administrator Profile Values page, individually search for each of the profile options listed above.
6. Do the following for each profile option:
 - a. In the Profile Values region, in the **Profile Level** drop-down list, ensure that **Site** is selected.
 - b. For the first three profile options, select **Yes** from the **Profile Value** drop-down list.
 - c. To change the value for the last two profile options, specify a value in the **Profile Value** field as explained in the table.
7. Click **Save**.

Validate Authentication with the AI Application

Now you must validate the authentication between Fusion Service and the AI application. This step ensures that all the previous steps are completed correctly.

Use the Inbound Email Configuration and Validation page in Fusion Service to complete the validation.

1. Open the Tasks panel tab and click **Search**.
2. Search for and click the **Manage Email Configuration, Registration, and Validation** task.
3. On the Inbound Email Configuration and Validation page, navigate to the Email Attribute Detection area.

4. In the Test Authentication area, click **Validate**.

This step validates whether you're able to generate the token. If the validation is successful, a tick mark indicates that the authentication service is working fine, and all settings are done correctly in Fusion Service.

Filter the Data for the Machine Learning Model

Duplicate and inactive requests negatively impact the quality of predictions.

If you want to clean up the data before it's fed to the Machine Learning API, create exclusion rules for data ingestion. These rules help to filter out the duplicate or inactive data, so they're excluded from being ingested by the ML model.

1. In the Setup and Maintenance work area, go to the following:

- o Offering: Service
- o Functional Area: Communication Channels
- o Task: Manage Email Configuration, Registration, and Validation

2. In the Inbound Email Configuration and Validation page, navigate to the SR Classification area.

3. In the Ingestion Exclusion Rule area, click **Add**.

4. In the **Object** field, select one of the following: Service Request, Internal Service Request, or HR Help Desk Request.

5. In the **Attribute** field, select the object attribute based on which you want to filter the data.

6. In the **Operator** field, select a suitable operator such as **Contains**, **Equals**, or **Is Not Null**, depending on the rule you're creating.

7. In the **Value** field, specify the value for the exclusion rule.

Here are some examples:

- o Requests created by messages from WhatsApp: Here, you could select the attribute as **Channel Type**, operator as **Equals** or **Contains**, and the value as **WhatsApp**.
- o Requests created by a user whose email pattern contains test: These requests aren't to be considered for training the ML model, because this would be test data.

8. Click **Save**.

Start the Data Ingestion and Model Training

Now, you can build a prediction model based on data pulled from closed and resolved requests created from inbound emails.

Before you start

Data from the last 6 months is used for the model training so the latest data is always included. You've the option to retrain the model on the latest data every 2 weeks. Every time the model is retrained, older data is discarded and new data is added.

Here's what to do

1. Sign in to Fusion Service with either the CRM SVC Attribute Detection Administrator or Adaptive Intelligence Applications Administrator role.
2. Click the **Tools** menu.
3. In the **Quick Actions** pane, click **AI Apps Administration**.
4. From the list of AI applications on the Application Administration page, click **Service Request Classification**.
The Configure SR Classification page is displayed, and the **Configuration** tab is selected by default.

5. In the Credentials to connect to Fusion Service area, specify the following details:
 - a. In the **User Name** field, enter the user name for your Fusion Service environment.
 - b. In the **Password** field, enter the password for your Fusion Service environment.
 - c. In the **Endpoint** field, enter the Oracle Fusion Service BIP Host name. For example: `https://<host_name>/xmlpserver`
 - d. Click **Save**.

Note: After the credentials are saved, you see a **Credentials Saved** message, and the **Activate** button in the Activate Tenant area is enabled. This indicates that the AI application can ingest the data from Fusion Service.

- e. To ignore the changes, click **Cancel**.
6. In the Activate Tenant area, do the following:
 - a. Click the **Model Retraining (Biweekly)** checkbox if you want the model to be retrained on the latest data every 2 weeks.
 - b. Click **Activate** to schedule the data ingestion and model training. If it's been activated before, click **Reactivate**.

Results:

The data ingestion and model training begins and the process runs in the background.

Validate the Setup

Validate the settings to see whether your model is working and that you're able to make a call to the prediction service.

1. Open the Tasks panel tab and click **Search**.
2. Search for and click the following task: **Manage Email Configuration, Registration, and Validation**.
3. On the Inbound Email Configuration and Validation page, navigate to the SR Classification area.
4. Navigate to the Test SR Classification API area.

Here, you simulate an inbound email.
5. Enter a subject in the **Subject** field.
6. In the **Plain Text Body** field, enter the text for the body of the email.
7. From the **Stripe Code** drop-down list, select the relevant stripe code.
8. From the **Business Unit** drop-down list, select the relevant business unit.
9. Enter a value in the **Language** field. For example, English. The language must be a one that is read from left to right.
10. Click **Validate**.
11. If the validation is successful, a tick mark is displayed.

This tick mark indicates that the process is working successfully. And it indicates that you're able to make a call to the SR Classification API that's deployed on the Adaptive Intelligence Apps platform. The Attribute Predictions area displays the attribute name, value, and accuracy for the prediction.

Note: If the data set of SRs in the Fusion Service database isn't very good, the accuracy and confidence of the prediction is low.

Test Service Request Classification

Test the Classification for New Requests

After completing the validations, you can conduct a test to see whether the category, and product are being stamped on the newly created service request (SR).

Before you start

Note: If the values of the profile options for prediction threshold are high, then it's possible that the category and product aren't stamped on the request. Therefore, it's a good idea to lower the threshold values using the profile options before sending the test email.

Here's what to do

1. Send a test email to the appropriate email channel that's synced with the predicted business unit.
2. Open the request and view the details in the **Summary** tab.
3. Verify whether the prediction attributes are present on the request.
4. After you complete testing, you can reset the values of the profile options for prediction threshold per your requirements.

Insights Dashboard to Evaluate Accuracy

About the Insights Dashboard

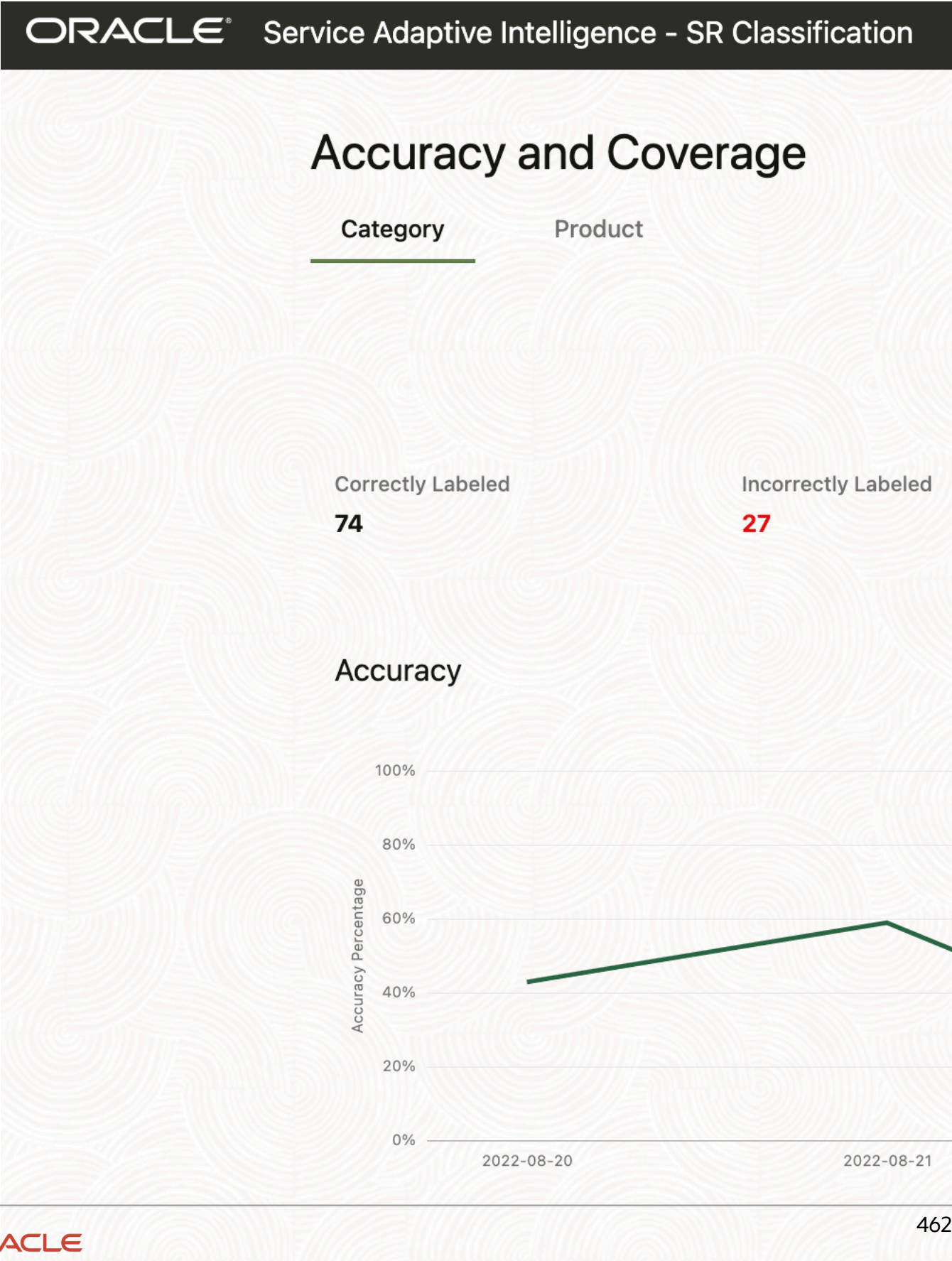
The insights Dashboard helps you understand prediction accuracy and coverage trends.

The Insights Dashboard provides analytical reporting to view the trend of accuracy and coverage metrics. You don't need to do anything to enable insights.

The Insights Dashboard provides the following categories of insights:

- **Activity and Coverage Insights.** Measured by the count of requests that are labeled, not labeled, correctly labeled, or incorrectly labeled. These insights trend

over time to give you a grasp of their variation across days, months, or years.



ORACLE

462

-
- **Accuracy Insights.** Measured by accuracy percentage. This is the count of correctly labeled requests expressed as a percentage of total count of requests labeled by AI. These

insights trend over time to give you a grasp of their variation across days, months, or years.

ORACLE® Service Adaptive Intelligence - SR Classification

Accuracy and Coverage

Category Product

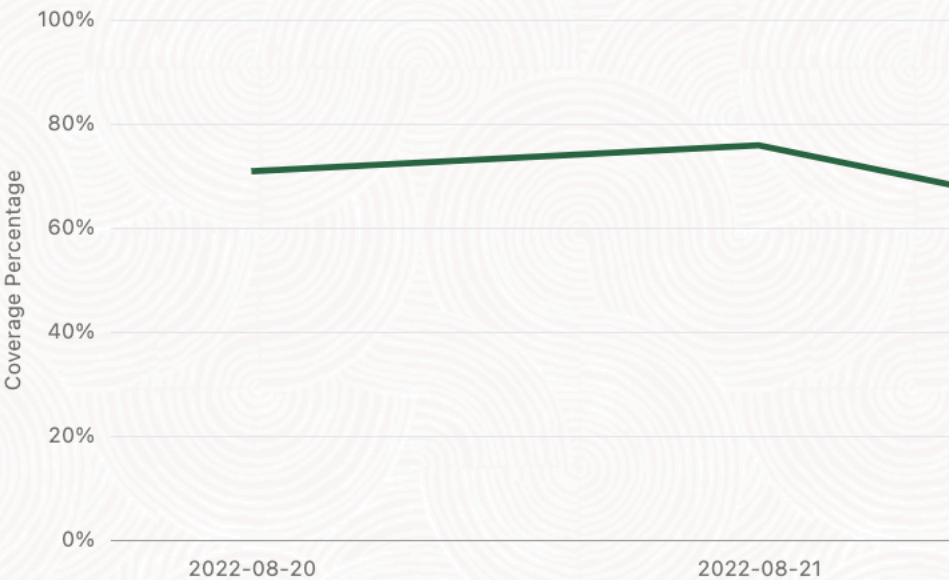
Correctly Labeled

74

Incorrectly Labeled

27

Coverage



- **Business Impact Insights.** Measured by the average time to resolve requests.

ORACLE® Service Adaptive Intelligence - SR Classification

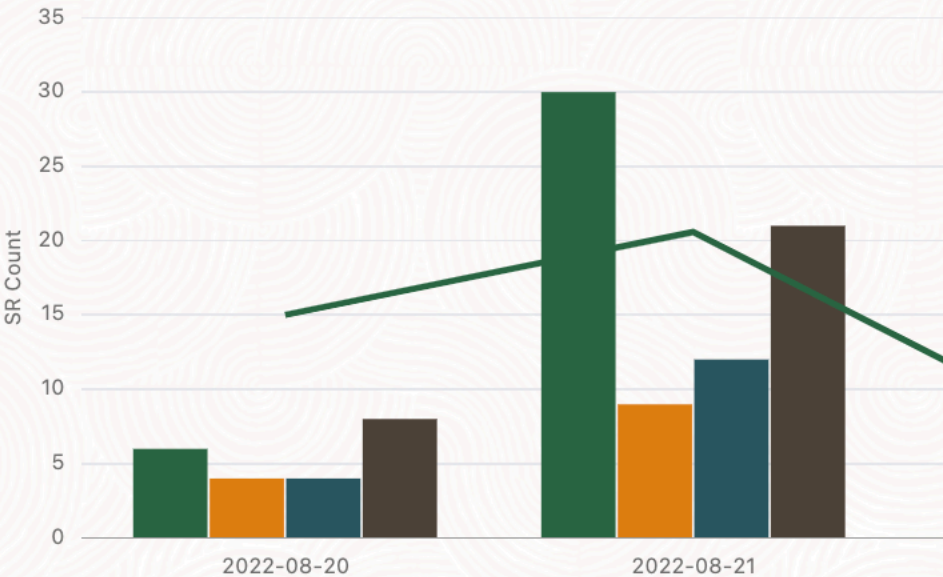
Accuracy and Coverage

Category Product

Correctly Labeled
74

Incorrectly Labeled
27

SR Count



You can generate a graph by selecting the required filters.

Here are some key points about the graph:

- For each of the insights, the x-axis of the graph shows the timeline.
- The graph shows two different views of the data:
 - In terms of actual volume of requests
 - In percentages, which are a proportional view
- The overall number of closed requests is divided into different categories, and each of them is shown separately in the first line.
- The graph shows the following details:
 - **Correctly Labeled:** Requests that were correctly labeled by the prediction service.
 - **Incorrectly Labeled:** Requests that were relabeled by agents because the prediction service yielded an incorrect label.
 - **Unlabeled:** Requests that remained unlabeled by the prediction service because the confidence was too low.
 - **Manually Labeled/Relabeled:** These requests are a combination of:
 - **Manually Labeled** - Requests with no initial label, or unlabeled requests
 - **Relabeled** - Incorrectly labeled requests
 - **Avg Time to Resolve:** The average time taken to resolve requests that were classified using the prediction service.

Note: For the Activity and Coverage insight:

- The y-axis of the graph shows the request count and cumulative percentage.
- The line graph for the correctly labeled requests is also displayed. This graph denotes the percentage of correctly labeled requests averaged over the whole time period.

Use the Insights Dashboard

This procedure covers using the insights dashboard.

1. Sign in to Fusion Service as a user with the following roles: CRM SVC Attribute Detection Administrator and Adaptive Intelligence Applications Administrator.
2. Click the **Tools** menu.
3. In the **Quick Actions** pane, click **AI Apps Administration**.
4. From the list of AI applications on the Application Administration page, click **Service Request Classification**.
5. On the Configure SR Classification page, click the **Insights** tab.
6. On the Accuracy and Coverage page, click the **Category** or **Product** tab, depending on which report you want to view.
7. Select one of the following options from the **Insights** drop-down list:
 - **SR Count:** Shows the count of SRs that are correctly labeled, incorrectly labeled, unlabeled, and manually labeled or relabeled.
 - **Accuracy:** Shows the accuracy percentage. This is the percentage of correctly labeled SRs out of the total number of SRs that were labeled by AI.
 - **Coverage:** Shows the coverage percentage. This is the percentage of correctly and incorrectly labeled SRs out of the total number of SRs that were labeled by AI.

8. Do one of the following:
 - Set **Exact Match** to **On** if you want the label to be counted as correct when there's an exact match between actual and predicted values for product or category.
 - Set **Exact Match** to **Off** if you want the label to be counted as correct when there's a match within the same hierarchy.
9. In the **Display by** drop-down list, select one of the following options:
 - **By Day**
 - **By Month**
 - **By Year**
10. In the first date picker, select the start date for the report.
11. In the second date picker, select the end date for the report.

28 Set Up Service Work Orders

Enable Work Orders

To expose the Redwood User Interface for Service, including Work Orders, you must enable the feature using the Opt-In page.

Refer to the Enable Service Center topic in Related Topics. Follow those steps to expose the **Work Orders (Next Generation)** tile, and the **My Open Work Orders** quick action on the home page.

For general information on work orders, refer to the Set Up Work Orders chapter in the Implementing Fusion Service guide.

Related Topics

- [How do I enable Service Center?](#)

Extend Work Orders UI Elements

You extend UI elements in your Redwood application using Visual Builder Studio.

You can make changes to the user interface, such as displaying custom fields in your work order forms, creating dynamic layouts for a form, or embedding custom content in a dynamic container page.

For information about extending your Work Order UI elements, including setting up Visual Builder Studio, refer the Extend Service Center chapter. The examples in that chapter pertain to the Service Request layouts, but the basics are the same.

29 Set Up Survey Integration

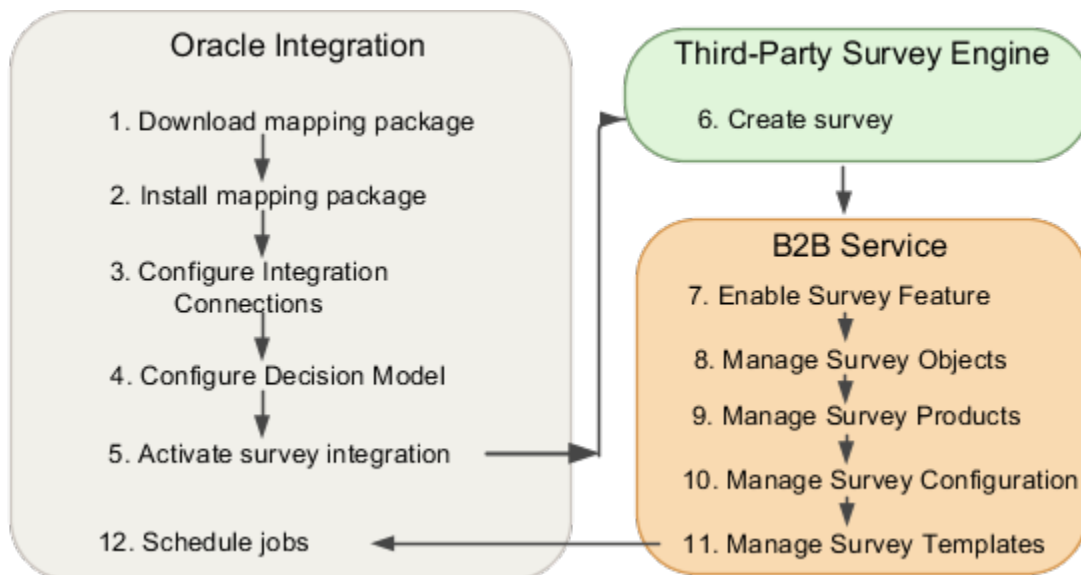
Can I use surveys?

The survey integration framework lets you start survey requests based on any object.

You can send surveys from your preferred third-party supplier to your customers based on your chosen rules. You can define who receives a survey, what survey should be sent, and when to send a survey.

The survey integration framework connects with your survey supplier using Oracle Integration Enterprise Edition.

The following graphic shows the set up steps you'll need to perform to integrate the survey framework.



Survey Requirements and Licensing

You must have subscriptions to the following cloud services to implement the Survey Integration Framework:

- Oracle Fusion Service 11.13.20.01 and higher
- Oracle Integration Enterprise Edition version 19.4.3.0.0
- Third-party survey product supplier such as a paid account with SurveyMonkey.

Use the Survey Framework with Oracle Integration

Oracle Integration synchronizes Oracle Fusion Service with your third-party survey supplier. Do the following tasks to set up the secure integration between customer-specific instances:

- Create a user with the Setup Survey privilege to connect to Oracle Integration
- Download the Oracle Integration Flows
- Configure Integration Connections

Create the Fusion Service Administrator User for Connecting to Oracle Integration

You must create a user that's used to connect from Oracle Integration to Fusion Service. Follow the steps in the Create Setup Users topic to create the user.

Grant the following roles to the new administrator user:

- Customer Service Manager (if using the Service Request object)
- Human Resource Help Desk Manager (if using HR Help Desk)
- SOA Operator
- Internal Help Desk Administration (if using Internal Help Desk)

Download and Save the Oracle Integration Mapping Package

The first step in setting up the Oracle Integration-based integration is to install the Oracle Integration mapping package from Oracle Marketplace.

1. Sign in to Oracle Marketplace.
2. Select **Products** from the drop-down list.
3. Enter Survey in the **Search** field.
4. Click **Go**.
5. Select the integration package Oracle Fusion Service Survey Integration Framework Sample Integration.
6. Click **Get App**.
7. Review and accept the Terms and Restrictions.
8. Click **Next**.

The package has two files with the following extensions:

- .dmn

- `.par`

9. Select the `.par` file.
10. Save the file to a local computer.

Note: The files in the package are specific examples for using SurveyMonkey as a third-party survey supplier. If you use another supplier, you can use the files as a reference example to create your own.

Related Topics

- [Oracle Cloud Marketplace](#)

Configure the Integration Connections

Complete the integration between Oracle Fusion Service and your third-party survey supplier in the following order. The steps in this procedure use SurveyMonkey as the example.

1. Import the package into Oracle Integration
2. Configure a connection to the Oracle Fusion Service Instance
3. Configure the Connection. In this example, the connection is configured for SurveyMonkey.

Note: If you're using SurveyMonkey, be sure to review the SurveyMonkey API information at the end of this chapter.

4. Configure the Survey Oracle Integration connection

Import the Package

The Oracle Integration administrator imports the package.

1. Sign in to Oracle Integration.
2. On the home page, click **Integrations**.
3. Click **Packages**.
4. On the Packages page, click the **Import** button.
5. On the Import Package window, browse for the `.par` file that you downloaded from Oracle Marketplace.
6. Click the **Import** button.
7. The `.par` file is now listed on the Packages page.

To view what's in the package, click the package in the **Name** column.

8. Click **Close**.

Note: The files in the package are specific examples for using SurveyMonkey as a third-party survey supplier. If you use another supplier, you can use the files as a reference example to create your own.

Configure the REST Connections

REST connections allow Oracle Integration to connect to the Survey Integration connection for your third-party survey supplier using API calls. Your supplier must have public APIs.

1. From the Oracle Integration home page, navigate to the Integrations page.

2. Click **Connections**.
3. Enter "Survey" in the **Search** field to narrow the search results.
4. On the Connections page, find and click the REST connection. In this example, we use SurveyMonkey. Find the REST connection **Survey Integration SurveyMonkey Connection**.
5. Click on Configure Connectivity and enter the value for Connection URL that's appropriate for your site. For example, `https://api.surveymonkey.com`.
6. Click **OK**.

7. On the Detail page, enter the required values for **Security**.

Note: You need your Client ID and Client Secret values to continue. These values are provided by your third-party supplier. In this example, SurveyMonkey. In the Settings tab in SurveyMonkey, change the **OAuth Redirect URL** to:

- `https://<Your OIC host>:443/icsapis/agent/oauth/callback`

This is the OIC URL you got from the browser. Here's how you get it.

- a.** Log in to Oracle Integration.
- b.** Copy and paste the OIC host from your browser. For example, it looks like `https://oicapps-oicdevqa.integration.ocp.oraclecloud.com`.

For the supplier SurveyMonkey, you must have the following scopes selected in SurveyMonkey under the settings tab:

- View Responses
- View Response details
- View Users
- View Surveys
- View Contacts
- View Collectors
- Create/modify Contacts
- Create/modify Collectors

Tip: You can find your SurveyMonkey Client ID and Client Secret values by viewing the details of your SurveyMonkey Private App at <https://developer.surveymonkey.com/apps>.

Continuing from the previous step, enter the required values for Security. The following table shows the values.

Required Security Property	Value
Security Policy	Select OAuth Custom Three Legged Flow from the drop-down list
Authorization Request	Enter the Authorization Request that's appropriate for your site. For example, <code>https://api.surveymonkey.com/oauth/authorize?response_type=code&redirect_uri=\${redirect_uri}&client_id=xxx</code>

Required Security Property	Value
	Where xxx is the Client ID you received from SurveyMonkey.
Access Token Request	Enter the Access Token Request that's appropriate for your site. For example, -X POST -H "Content-Type: application/x-www-form-urlencoded" -d "code=\${auth_code}&redirect_uri=\${redirect_uri}&grant_type=authorization_code&client_id=xxx&client_secret=yyy" https://api.surveymonkey.com/oauth/token Where xxx is Client ID and yyy is Client Secret

8. Click **Provide Consent**. For more information, see the FAQs in this chapter.

Before continuing, make sure you receive a success message (access allowed).

9. When you're prompted, enter your Oracle Integration User Name and Password.

Note: If you encounter the error "The authorization request failed: Invalid redirect uri," copy the redirect_uri parameter value from the browser URL and update the same OAuth Redirect URL as you did in the previous note.

10. Enter your SurveyMonkey User Name and Password.
11. Click **Save** on the Provide Consent window.
12. Click **Test**.

You see a message saying the test was successful.

13. Click **Save**.

You see a window listing the integrations that are using the connection.

14. Click **Save** on window.
15. Click **Close** on the Survey Integration Oracle Integration Connection page.

CAUTION: For the integration to access the API for specific functionality of the survey, you must have active scope requirements in your SurveyMonkey Private App.

Configure the Oracle Fusion Service Instance Connection

This connection is used by Oracle Integration to get survey details from Fusion Service

1. On the Oracle Integration home page, click **Connections**.
2. On the Connections page, find Survey Integration Engagement Cloud Connection.
3. Go to the Survey Integration Engagement Cloud Connection Detail page.

4. In the Connection Properties window, enter the values as shown in the following table.

The values in the tables in the following steps are located in the email you receive when Oracle Integration is provisioned.

Property Name	Property Value
OSC Service Catalog	Enter the service catalog URL for your Oracle Fusion Service instance. For example: <code>https://<CRM domain WSDL URL host>/fndAppCoreServices/ServiceCatalogService?wsdl</code>
OSC Event Catalog URL	Optionally, enter the event catalog URL for your Oracle Fusion Service instance. For example: <code>https://<CRM domain host>/soa-infra</code>
Interface Catalog URL	Enter the interface catalog URL for your Oracle Fusion Service instance. For example: <code>https://<CRM domain host>/helpPortalApi/otherResources/latest/interfaceCatalogs</code>

5. In the Security region, enter the values as shown in the following table.

Property Name	Property Value
User Name	<FUSION SERVICE USER NAME>
Password	Enter the password.
Confirm Password	Reenter the password.

6. Click the **Test** button.

It may take some time for the user <FUSION SERVICE USER NAME> to synchronize with the directory.

A message appears saying the test was successful.

7. Click **Save**.

A window appears listing the integrations that are using the connection.

8. Click **Save**.

9. Click **Close** on the Survey Integration Oracle Integration Connection page.

Configure the Survey Integration Oracle Integration Connection

This connection is used by Fusion Service to connect to Oracle Integration.

1. On the Oracle Integration home page, click **Connections**.
2. On the Connections page, find Survey OIC Connection.
3. Click **Survey OIC Connection**.

4. Enter the values as shown in the following table.

Connection Property	Property Value
Connection Type	REST API Base URL
Connection URL	Enter the Oracle Integration host name URL shown in your browser's address bar. For example: https://oicapps-oicdevqa.integration.ocp.oraclecloud.com

5. In the Security region, enter your Oracle Integration User Name and Password values.
6. Verify your password is correct and click **Test**.
A message appears saying the test was successful.
7. Click **Save**.
A window appears listing the integrations that are using the connection.
8. Click **Save** on the window.
9. Click **Close** on the Survey Integration Oracle Integration Connection page.

Related Topics

- [Creating Decisions](#)

Configure the Decision Model

Configure the decision model to determine when a survey is sent.

There are two files in the integration package you downloaded in an earlier step. The `.par` file, which you already used to import the integration package, and the other is a file with the extension `.dmn`.

You use the `.dmn` file to configure the PCS decision model. The file is an example of a decision model that uses the service request object. If you don't have the same conditions, use the file as a reference and make changes that suit your needs.

If you want to have multiple decision models for the same object, download the file once and import it multiple times to keep the same type definition. For example, for predefined SRs you can import the `.dmn` file multiple times with a different name. Otherwise, you must manually copy over all the fields of the SR to each new decision model.

Import and Create the Decision Model

Here's how to import the decision model.

1. On the Oracle Integration home page, go to **Decision Models**.
2. Click **Create**.
3. Click **Import a Decision Model**.
4. Browse for the `.dmn` file that you downloaded with the integration package.
5. Enter the **Name**, **Space**, and **Description**.
6. Make sure the **Open Immediately** check box is checked.

Once you import the decision model, click the model to begin configuring the rules for the survey.

7. Activate the decision model.

For more information about creating and testing decision models, refer to the Creating Decisions chapter of the Using Oracle Process Cloud Service guide.

Related Topics

- [Creating Decisions](#)

Activate the Survey Integration Flow

After importing the integration package and successfully connecting Oracle Fusion Service, you can now activate the integration flow.

Activate the Integration Flow

Follow these steps to activate the integration flow.

1. Sign in to Oracle Integration.
2. On the home page, click **Integrations**.
3. Click **Packages**.
4. Search for the **EC_SAMPLE_SURVEY_INTEGRATION** package.
5. Click the package name.

A window appears that lists the integrations in the package.

6. You must activate each integration individually, so take note of the names of each integration in the package.
7. Click **Integrations** in the menu.
8. Find each integration in the package, and click the **Activate** icon in the status column.

For example:

- Click the **Activate** icon in the Status column for your integrations package.
- Click **Activate** on the confirmation dialog window.

CAUTION: Oracle doesn't recommended enabling tracing when activating the integration flow in production. If for some reason you enable tracing, you should not select the include payload option.

- Make sure the flow was activated successfully.
9. Verify that the integration synchronization is working.

Note Your Integration Endpoints

After activating and testing your integration flows, you must note your activation endpoints. You need your endpoints later when you configure your third-party supplier's survey in Fusion Service.

Endpoints connect your surveys from your survey supplier to Fusion Service so when you're creating a survey template, all your surveys are visible in Fusion Service.

For example, to find the endpoints using SurveyMonkey as your supplier::

1. Navigate to Integrations in Oracle Integration.
2. Click the **Link** icon in the **Status** column of the **Get SurveyMonkey Surveys** integration.
3. Click the **Metadata URL** link.

A new tab that contains your Endpoint URL opens. Note the Endpoint URL for later when you set up survey configuration in Fusion Service.

Create Your Survey

Once you have completed the previous steps, continue setting up Survey in Fusion Service, by creating a survey using your third-party survey supplier. Your survey is created, maintained, and sent by that supplier. To create your survey, refer to your supplier's documentation.

Follow the steps in your supplier's documentation for Summary, Design, Preview and Score, and Collect Responses. Optionally, follow the steps for Analyze and Present Results.

Note: If you're using SurveyMonkey as your supplier, Create a Collector and Create a Message are required steps. When you create a Collector, remember that Fusion Service only supports an email channel for collecting survey responses.

Service Request Attributes Used in Templates

If you're using SRs as your survey object, you can use any SR attribute as a part of your survey's email template.

For SR's add attributes to your template by adding the attribute inside double brackets.

Here's a sample email template using the two most common SR attributes from Fusion Service.

```
Hi {{PrimaryContactPartyName}},
```

```
We're conducting a survey and your input is appreciated. Click the button below to start the survey. Your SR: {{SrNumber}}
```

You can get the SR attribute values from the API Name in Application Composer or from REST API.

Most predefined SR fields are already mapped. Here's how you can see the previously mapped fields:

For this case, we will continue using SurveyMonkey as the example.

1. Log in to Oracle Integration.
2. Search for **SurveyMonkey Survey Request** Job.
3. Select the job.
4. Find the **Map to TestQualifySR** in the flow.

Note: The icon next to Map to TestQualifySR is actually the link to the Decision Model in Process Cloud.

5. Click **Map to TestQualifySR** and click **Edit**.

The mapping appears.

Attributes that have a line mapping from the \$CurrentSR source are pre-mapped. You can add those attributes to your template by framing them with double brackets as shown in the sample.

To use attributes in your template that aren't pre-mapped, or a non-predefined field that you have created, follow the steps in the Create Mappings topic.

Related Topics

- [Survey Monkey Developer Documentation](#)
- [Create Mappings](#)

Supported Survey Question Types

When creating your survey questions in the third-party survey supplier tool, you can base them on certain question types.

Here are the types of surveys that are supported with the survey framework:

- Text
- Check box (single select)
- Multiple select check box
- Slider
- Drop down
- Multiple line text
- Option

Enable Surveys in Fusion Service

Follow these steps to enable and set up the Survey feature in Fusion Service.

Note: You must be a Fusion Service Administrator with the Setup Survey privilege and all prior tasks in this chapter must be completed before you can proceed.

To opt in to the Survey feature:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Change Feature Opt in link
2. Select **Enable** for Surveys.
3. Click **Done**.

The Survey feature now appears in the Functional Area of the Setup Service page.

Manage Survey Products

Here's how you add your third-party survey supplier to the integration.

Note: The predefined product is SurveyMonkey. If that's your third-party survey supplier, there's no need to follow these steps.

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Feature: Surveys
 - o Task: Manage Survey Products
2. In the Lookup Codes section, click the **Add** icon.
3. In the Lookup Code field, enter a lookup code for your supplier. For example, CUST_SUPPLIER_NAME.
4. Enter the name of your supplier in the **Meaning** field.
5. Enter a description in the **Description** field.
6. Click **Save and Close**.

Manage Survey Objects

In this chapter, we use service requests as the object used for the Survey Framework feature. You can use different objects for surveys.

If you're also using the service request object, there's no need to follow these steps. Otherwise, here's how you add survey objects.

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Feature: Surveys
 - o Task: Manage Survey Objects
2. In the Lookup Codes section, click the **Add** icon.
3. In the Lookup Code field, enter a lookup code for the object. The predefined object is service requests. If you're using a different object, enter something like CUST_SVC_ACCOUNT.
4. Enter the name of the object in the **Meaning** field. For example, Service Request.
5. Enter a description in the **Description** field. For example, The lookup value for the Service Request.

This object is used later when you create a survey template using the Manage Survey Templates task.

6. Click **Save and Close**.

Manage Survey Configurations

You can now configure Survey in Fusion Service. This configuration is used to map the third-party survey supplier (in this example SurveyMonkey), the Oracle Integration URL, User Name, and Password to Fusion Service.

Follow these steps to enable and set up Survey:

SurveyMonkey is used in this example.

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service
 - o Feature: Surveys
 - o Task: Manage Survey Configurations
2. In the Survey Products region, click **Add**.
3. Select your supplier from the **Survey Product** drop down list. This is the supplier you created in the Manage Survey Products task. In this example, we are using SurveyMonkey.
4. Enter a name for the configuration.

The configuration name is used later when you create the survey template.

5. Enter the values as shown in the following table:

Field	Value
Name	Enter a name for the configuration.
Description	Enter a description for the configuration.
Integration User	Enter your Oracle Integration User Name
Integration Password	Enter your Oracle Integration Password
Integration Endpoint URL	Paste the endpoint URL that you noted earlier. For example: <code>https://oicapps-oicdevqa.integration.ocp.oraclecloud.com:443/ic/api/integration/v1/flows/rest/GET_SURVEYMONKEY_SURVEYS/1.0/surveys? [Page-value] &PerPage=[PerPage-value] &Name=[Name-value]</code>

Note: When you paste the URL, remove everything starting from (and including) the question mark in the Endpoint URL.

6. Click **Save and Close**.

Manage Survey Templates Task

Survey templates are used to map your survey with decision rules you use to send the survey.

To create and manage survey templates:

In this example, we use SurveyMonkey.

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Feature: Surveys
 - Task: Manage Survey Templates (to create and manage general SR survey templates)

Or

 - Task: Manage Survey Templates for HR Help Desk (to create and manage survey templates for HR Help Desk surveys)
2. To create a template, click **Add**.
3. Enter a **Name** and **Description** for the template.
4. Select the **SurveyMonkey Configuration Name** from the drop-down list.

This name comes from the configuration you created in the Manage Survey Configurations task.

5. Enter a **Start Date** for the survey.
6. Optionally, enter an **End Date**.
7. Select **Service Request** from the **Object Type** drop-down list.

If you're using an object different from a service request, here's where you choose it. For example, in the Manage Survey Objects task you completed earlier, we used the example of entering CUST_SVC_ACCOUNT as the object type.

8. Select the **External Survey Name** from the drop-down list.

The list is of all the surveys you have in SurveyMonkey.

9. Enter the **External Survey URL**.

This is the direct link to your third-party survey.

10. Select a **Decision Model Space**.

The decision model space is the one you created earlier.

11. Select a **Decision Model Name**.

The decision model name is the one you created earlier.

12. Click **Save and Close**.

Schedule Jobs for Sending Surveys and Collecting Survey Responses

In this task, you schedule and manage jobs. A job identifies all the service requests that are applied to decision rules that you configure.

For example, send a survey to the primary contact of all service requests that have been closed since the last job run. In this example, SurveyMonkey is used as the sample job.

1. From the Integrations page in Oracle Integration, search for the survey job **SurveyMonkey Optimized Survey Request Job**.
2. Select **Schedule** from the hamburger menu under **Status**.
3. On the Schedule page, click the **Edit** icon.

You can edit the type, frequency, effective dates, and so on.

Note: To schedule a survey, your survey must be active in Fusion Service.

4. Click **Start Schedule**.
5. Click **Start Schedule** on the Scheduled Run window.

Note: Scheduled jobs can use a lot of resources. Consider scheduling your jobs for off-hours, or only once or twice a day. If you're using SurveyMonkey, activate the schedule for both of the following jobs using the steps in this procedure.

- **SurveyMonkey Optimized Survey Request Job** locates the records that match the rules and sends the requests to the third-party supplier.
- **SurveyMonkey Survey Response Sync Job** goes out to the third-party supplier, gets the survey responses, and brings them into Fusion Service so BI reports can be run to view the data.

Related Topics

- [Scheduling Integration Runs](#)

Verify the Survey Integration is Working

You can now test the integration flows.

Verify the Integration is Working

Follow these steps to verify that the integration is working. In this example, we verify the integration for the third-party supplier SurveyMonkey.

1. If you haven't already done so, create a test survey in SurveyMonkey with a corresponding message.
For more information, see the topic [Create your SurveyMonkey Survey](#).

2. Create a test rule. Refer to the topic in this chapter about configuring the Process Cloud decision model.
3. Go to Manage Survey Templates task and create a test survey.
4. Make sure that:
 - The SurveyMonkey survey is visible under External Survey.
 - The Decision Model Space and Decision Model Name display the test rule.
5. You can test the survey by creating a test Primary Contact and test SR that satisfies the rule you created.
6. Wait for the job to run.
7. You should receive the test survey in the test primary contact's inbox.

How to Include an Administrator-Defined Attribute in the Decision Model

If you have an administrator-defined attribute in your service request and you want to include it in your decision model follow the steps in this topic. This example uses service requests as the object.

For example, you may have created an administrator-defined attribute to capture the level of service based on the Account on the service request such as, Gold, Silver, and Bronze. Let's say you want to include that field to help identify only the Gold Accounts that have a closed SR and send them a different survey than Silver and Bronze. In order to do this, you must expose the administrator-defined attribute in your decision model.

Add the Administrator-Defined Attribute to Your Decision Model

Once you add your administrator-defined attribute to your service request object using Application Composer, here's how you can add the attribute to your decision model.

1. Navigate to Oracle Integration > Processes > Decision Models.
2. Select the Decision Model you want to edit.
3. Expand Service Request in the **Type Definition** region.
4. Edit the **Type Definition** to include the new fields by clicking the menu and selecting **Edit**.
5. In the **Edit Type Definition** dialog box, click the **Add** icon in the Define Type Attributes section.
6. In the **Add Component** dialog box, enter the Name of the administrator-defined attribute.
7. Use the drop-down list to select the **Mode**.
8. Click Close.

You can now include the attribute in the decision model.

9. Click **Activate**.
10. On the **Create Snapshot and activate** dialog window, click **Activate**.

Note: There may be characters that were appended to the decision model Name. Make sure the Name is an exact match to the name of your decision model in the Manage Survey Configurations task in Functional Setup Manager.

Don't forget to test the decision model to make sure it works.

Change the Integration to Include the Administrator-Defined Field

Now that you have added the attribute to the decision model, you must change the integration so that the new field can be used in the decision model.

1. Navigate to Oracle Integration > Integrations.
2. Find the integration you want to edit.

If the job is scheduled, you must pause or stop the job before you continue.

3. Now, deactivate the job by clicking the **Deactivate** icon.
4. Click the job name to edit.

First, you update the initial query of SRs to pull the new attribute from Fusion Service.

5. On the Scheduled Orchestration page, navigate to `Invoke GetSRPage` using `Survey Integration Engagement Cloud Connection` connection of type `Oracle Engagement Cloud`.

`Invoke GetSRPage...` is where the fields of the SR are being picked up.

6. Click the `Invoke GetSRPage...` Three buttons appear and then click the **Regenerate** link.

This regenerates the SR fields to include the administrator-defined field.

7. Click **Regenerate** on the Regenerate Artifacts dialog box.

Next, you update the mapping to include the administrator-defined field in the decision model.

Tip: In between actions, it's a good idea to click **Save**.

8. On the Scheduled Orchestration page, navigate to the `Invoke TestQualifySR` using `Survey Integration OIC Connection` connection of type `REST`.

`Invoke TestQualifySR...` is the actual call to the decision model.

9. Click the icon and select **Edit**.

10. On the Configure REST Endpoint dialog window, select the **Request** tab.

11. In the Select the request payload format section, click the `<<<inline>>>` link.

12. Scroll to the end.

13. Use valid JSON script. Add a comma and press enter on your keyboard.

14. Add the name of the administrator-defined field in quotes.

15. Add a colon after the quotes.

16. After the colon, enter an example value.

- If the value is true or false, type either true or false.
- If the value is a number, you can type any number.
- If the value is text or date time, you can type any value in quotes. It doesn't matter what you type inside the quotes.

The following table shows examples of the last few lines of the sample JSON payload.

JSON Syntax	Sample (last few lines of the sample JSON payload)
Original:	<pre>"CaseNumber" : "ABCDEFGHI", "CaseId" : "ABCDEFGHIJK", "PrimaryContactPersonId" : "ABCDEFGHIJKLMNOPQRSTUVWXYZ" }</pre>

JSON Syntax	Sample (last few lines of the sample JSON payload)
	}
Adding a true/false field:	<pre>"CaseNumber" : "ABCDEFGH", "CaseId" : "ABCDEFGHIJK", "PrimaryContactPersonId" : "ABCDEFGHIJKLMNPOQRSTUVWXYZ", "MyBooleanField": true } }</pre>
Adding a text field:	<pre>"CaseNumber" : "ABCDEFGH", "CaseId" : "ABCDEFGHIJK", "PrimaryContactPersonId" : "ABCDEFGHIJKLMNPOQRSTUVWXYZ", "MyTextField": "blahblahblah" } }</pre>
Adding a date/time field:	<pre>"CaseNumber" : "ABCDEFGH", "CaseId" : "ABCDEFGHIJK", "PrimaryContactPersonId" : "ABCDEFGHIJKLMNPOQRSTUVWXYZ", "MyDateField": "2020-06-01" } }</pre>
Adding a number field:	<pre>"CaseNumber" : "ABCDEFGH", "CaseId" : "ABCDEFGHIJK", "PrimaryContactPersonId" : "ABCDEFGHIJKLMNPOQRSTUVWXYZ", "MyNumberField": 1234 } }</pre>

17. If you're adding multiple fields, repeat the last steps. Add a comma after the last field, press enter, and so on.
18. Click **OK**.
19. Click the **Summary** tab.
20. Click **Done**.
21. Click **Update** on the Update Configuration dialog box.

Tip: Remember to save between actions.

Now, you can edit the mapping.

22. On the Scheduled Orchestration page, navigate to **Map from Schedule to TestQualifySR**.
The icon for **Map from Schedule to TestQualifySR** is the mapping from the SR to the decision model.
23. Click the icon and select **Edit**.
The Map page opens.
24. In the Target panel, expand the items until you see Service Request.
The Target panel represents the decision model.
25. Scroll through the Target list to find the new administrator-defined field.
26. In the Sources panel, expand **\$CurrentSR**.
27. Scroll down the **\$CurrentSR** list until you find the same administrator-defined field.
28. Click the administrator-defined field in the Sources panel and drag it to the same field in the Target panel.

This action creates a line linking the field.

29. After mapping the new field from Source to Target, expand the field (on the Target side of the page) to expose a nil row.
30. Right click on the word `nil` and select **Create Target Node**.
An **Expression for: nil** dialog window opens.
31. Click the **Switch to Developer View icon** on the **Expression for: nil** dialog window.
32. In the **Expression for: nil** dialog window, enter `true()`.
33. To save your changes, click the **Check mark icon** on the **Expression for: nil** dialog window.
An icon is now displayed in the Mapping Canvas panel that shows a mapping to the word nil.
34. Click **Validate**.
35. You see a message saying that mapping is valid and ready to use.
36. Click **Close**.
37. Click **Save** on the Scheduled Orchestration page.
38. Click **Close**.
The final step is to reactivate the integration.
39. Click the **Activate** icon in the Status column of the integration row.
40. On the Activate Integration dialog window, make sure Enable Tracing is deselected and click **Activate and Schedule**.

Related Topics

- [Creating Decisions](#)
- [Working with Decision Models](#)
- [Adding and Ordering Decisions](#)

Export and Import the Functional Setup Data for Surveys

If you create the functional setup for the Survey feature in your test environment, you can export the data and import it into your production environment.

You can do this using either the implementation method based on an offering, or from an implementation project. For more information about how to export and import your setup data, see the links at the end of this topic.

Whichever method you use, when the import is complete, you must reconnect the integration by reentering your Endpoint Password and URL (if the URL is different from your test environment).

Here's what to do after your setup data import is complete:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Feature: Surveys
 - Task: Manage Survey Configurations
2. Select the row for the survey configuration.
3. In the Survey Product API Properties region, enter the **Survey Integration User**.

4. Reenter the **Survey Integration Password**.
5. If you have multiple integrations, verify the URL in the **Survey Endpoint URL** field is correct.
For more information about configuration, see the Manage Survey Configuration topic earlier in this chapter.
6. Verify the survey integration is working.
For more information about verifying the integration, see the Verify the Survey Integration is Working topic earlier in this chapter.

Note: Be sure to stop your jobs in the test environment before you start the production integration connection. After you verify the integration is working in the production environment, start your jobs again.

Related Topics

- [Setup Data Export and Import Using Implementation Project](#)
- [Export Setup Data Using Implementation Project](#)
- [Setup Data Export and Import Using an Offering or a Functional Area](#)

How to Change an Active Decision Model

To change the rules for an active decision model where a job is already running, there are a series of additional steps you must take.

The steps in this example change a current, active decision model where a job is already running. In this example, the decision is set to send surveys when the SR status is Resolved. We want to change it so that surveys are sent when the SR status is Closed.

To do this, you must deactivate and delete the active decision snapshot and then create and activate it again. Here's how:

Pause the Schedule

Before you start, pause the scheduled job for the active rule.

1. Sign in to OIC (Integration Cloud).
2. Search for the survey integration and click the **Pause Schedule** icon on the active job.

Change the Rule

Now, let's change the rule.

1. On the Integration page, go to **Decision Models** and select "EC_SAMPLE_SURVEY_DECISION_MODEL".
2. Select the Decision **SR Status is Resolved** and change the if clause to `ServiceRequest.StatusTypeCd="ORA_SVC_CLOSED"`.
3. Click **Save**.
4. Click **Snapshots** from the Menu bar.
5. In the Manage Snapshots window, click the hamburger menu and select **Deactivate**.
6. Click **Yes** on the **Deactivate** window.
7. Click the hamburger menu and select **Delete**.
8. Click **Yes** on the **Delete Snapshot** window.

9. Click the **Activate** on the menu bar.
10. Add the rule again with the same name as the one you deleted.

Note: The application shows the previous name, but is appended with "-s1". The next two steps help you make sure the new snapshot name and version number match the old name and version number.
11. Make sure the snapshot **Name** is exactly the same as the old name.
12. Enter the same **Version** number as the original snapshot.
13. Check the **Override** check box to override the old version.
14. Click **Activate**.
15. Verify the snapshot is created by clicking the hamburger icon of the **Snapshots** menu and click **View**.
You should see the new snapshot.

Resume the Schedule

Don't forget to resume the job. Follow the steps at the start of this procedure, but click **Resume Schedule** instead of **Pause Schedule**.

What are the prerequisites for survey integration using SurveyMonkey?

Throughout this chapter, we used SurveyMonkey as the example for the third-party survey supplier. If you're using SurveyMonkey, you must follow these steps prior to using the Survey Integration Framework with SurveyMonkey.

Create a paid SurveyMonkey account and create a Private App

You can create a free account and then upgrade to a paid plan, or upgrade your existing free account.

1. Go to <https://developer.surveymonkey.com/apps/> and create an account.
2. Log in to SurveyMonkey.
3. Go to My Apps and create a new Private App.
4. Review the credentials and select the scope items used in your scenario. You can change this at any time.

Check your current API call limits

SurveyMonkey imposes a daily limit on the number of API calls. Therefore, there is a limit to the number of surveys you can send in a day. SurveyMonkey's documentation explains API call maximums. You must license your SurveyMonkey account appropriately for all connections over their defined limits, so make sure you have purchased enough API calls.

Note: SurveyMonkey monitors your API calls and if you exceed your limit, you're notified through email to upgrade your plan.

To check your API call limits:

1. Log in to SurveyMonkey.
2. Find your App and click the App name.
3. Click the Overview tab to see your API limits.

-
4. To adjust your API call limits, send an email to: **api-support@surveymonkey.com**

FAQs About Survey Integration

Frequently Asked Questions (FAQs) for Survey Integration

This section contains some frequently asked questions (FAQs) when setting up and maintaining survey integration. Scroll or search to find answers to common questions.

What happens if there are failures with Oracle Integration jobs?

You can investigate job failures in Oracle Integration. Sign in your Oracle Integration account and on the home page click Monitoring. On the integration tab, go to Runs. You can see your survey job run log.

What happens if there is a failure testing one of the Oracle Integration connections?

Check to make sure all the credentials you entered are correct. For example, the endpoint connection and your User ID and Password.

What happens if there is a failure with the Oracle Integration activation?

If there is a failure with the Oracle Integration activation, a link for the log that shows the reason for the failure is provided.

Isn't there a built-in survey tool in Fusion Service?

No. You must use a third-party survey supplier.

Can I use the survey integration feature without a paid SurveyMonkey account?

No. You must have a paid account with SurveyMonkey with the appropriate number of allotted API calls in your plan.

Where can I find a sample survey package for Fusion Service?

You can find the sample survey framework integration package on [Oracle Marketplace](#).

Which survey template is sent if an SR qualifies for more than one decision model?

The survey is sent for the most recently updated survey template in Fusion Service.

Can I have more than one decision model?

Yes. You can have as many decision models as you want.

Can I have more than one Friendly Enough expression in a decision model?

You can have as many Friendly Enough expressions as you want.

Can a decision model be based on any object?

Yes. For example, you can base the decision model on the object Active Account. You must make sure the object is passed to the decision model.

How can I deactivate a decision model?

In the Oracle Integration decision model, click on Snapshots (next to Activate) and then find the snapshot you want to deactivate. Click the hamburger menu on the snapshot and click Deactivate.

How can I map a nonstandard attribute to my email template?

To use attributes in your email template that aren't predefined, follow the steps to [Create Mappings](#) in the Oracle Integration documentation.

Which SR attributes are mapped ready to use?

Most predefined SR fields are already mapped. You can find pre-mapped fields in Oracle Integration.

Can I have multiple survey templates when I set up surveys in Fusion Service?

Yes. Each template corresponds to one survey within your third-party survey supplier.

What happens if I don't specify an end date on a survey template?

Your survey is sent as long as there are SRs that match the decision model rule.

What happens if a job doesn't run?

If a job doesn't run, or if you think there was an error, go to Track Runs in Oracle Integration. To access Track Runs, click Runs from the Integrations menu in Oracle Integration.

What happens if a job fails?

Both the request job and response job have Schedule Parameters that determine the time frame to consider for sending surveys and recording responses.

- Request Job: `startSurveyFromDate`: If the value of `startSurveyFromDate` is X, then the job will only test SRs that were resolved after X, where X is a date and time.
- Response Job: `LastStartTime`: If `LastResponseTime` is Y, then it only pulls responses to the survey submitted after Y.

If a job fails or errors occur when sending surveys or collecting responses, you can submit an ad hoc run outside of the normal schedule to cover the time frame in which the error occurred.

- Request Job: If you find that some surveys weren't sent since 1:00pm on March 14th, you can provide this date and time to an ad hoc run (2020-03-14T13:00:00). Then you can restart the normal schedule.
- Response Job: If you find that some error occurred and new survey responses weren't collected since 1:00pm on March 14th, you can provide this date and time to an ad hoc run (2020-03-14T13:00:00). Then you can restart the normal schedule.

Note: Date and time must be in ISO 8601 format. For example, (2020-03-14T13:00:00).

How can I extend the survey object?

You can extend two survey objects using Application Composer:

- Survey Response
- Survey Request

Note: There are no out of the box UI pages for these objects. To view the extended objects, you must build your own VB pages or use REST calls to access the extended objects.

Are there pre-built BI reports for Surveys?

Yes. Survey integration framework comes with pre-built, role-based, best-practice reports. The reports are located in the BI Catalog area at: Shared/Service/Embedded Content/Customer Service Manage/Survey Analysis. For a list of pre-built reports visit: Review prebuilt analytics for Fusion Service.

In addition, you can perform ad hoc analyses of your survey data using purpose-built subject areas. For more information, see the Service - CRM Survey Requests Real Time topic in the Subject for Transactional Business Intelligence guide.

Related Topics

- [CRM Survey Requests Summary](#)
- [Service-Reports.xls](#)

30 Service Analytics

How do I view analyses using OTBI?

Oracle Transactional Business Intelligence (OTBI) is a real time, self-service reporting solution bundled with your application.

OTBI 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 extent of your service-related business operations.

Here's how you can view the analyses:

- To access BI Catalog, select **Navigator > Tools > Reports and Analytics**. The Reports and Analytics page is displayed.
 - Click the **Hierarchical Selector >>** icon and select **My Folders** or **Shared Folders** from the menu. Alternately, you can click the **Browse Catalog** button. The BI Catalog is displayed, where you can view your personal and shared analyses. You can also create your own analyses based on your business requirements using subject areas. You need to have the necessary privilege for authoring analyses.
- To access the Analytics page, from the Home page, click the Service group icon, and then click the Analytics icon. The Analytics page is displayed. You can search for the analyses available in BI Catalog, and mark your favorites. These favorites stay on the Analytics page as long as they remain favorites. This page also shows the analytics recently viewed by the signed in user.

Service Analytics Roles

The following job roles enable to view analyses:

The Service analyses are visible for the customer service representative and customer service manager job roles. Analyses and the underlying data is secured through a set of delivered OTBI transaction analysis duty roles.

The user with a BI Administrator role 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 an existing catalog group.
- Assign permissions to a catalog object.

The OTBI transactional analysis duty roles for your service application are described in detail in the related topic "About Security Roles".

Service Dashboard

Service managers and agents can monitor their team and individual service performance using the role based, interactive service dashboard.

To access the dashboard, go to the **Service** menu on the home page and click the **Service Dashboard** Quick Action list.

The dashboard shows a series of prebuilt infolets, which display metrics/key performance indicators (KPI). An infolet may also display an analytic like a bar graph or a pie chart. Upon drilling into the infolet, a detailed report in tabular format is shown. For example, the **Critical Service Requests** infolet shows the list of individual SRs with their Reference Number, Title, Status, Assignee, and other relevant information. Users can navigate to individual records of the business objects displayed on a detailed report. For example, by clicking on the SR Reference Number one can navigate to the SR detail page and perform any updates to the SR and navigate back to the dashboard to resume analysis.

The service dashboard is can also be personalized to suit individual preferences. One can choose to hide/show the out of box infolets. Use the hide/show icon on the infolet to do so. As an end user (manager or agent) you can also change the order of display of the infolets. You can do this by clicking on the infolet with your mouse and dragging it to desired position on the dashboard. Once you do this, a **Save** button appears, after which you can choose to save the new layout or cancel it. Note that all personalization changes are user and browser specific.

Custom Infolet

You can add your own infolets to the dashboard. To do this, click on the menu button (ellipsis ...) at the top right corner of the dashboard and click **Add Infolet**. From the panel that appears on the right, navigate to the location of the Tile report in the BI catalog.

Here's a sample folder: Shared Folders/Custom/Enterprise Support/Strategic Service Requests/Tile: Escalated SRs.

For a BI report to be added as an infolet, its name should be in the format Tile: <<name of report>>. Click the report to be added as an infolet and click Next. At this point you can configure the infolet and to do so you are presented with a series of options to format . After selecting the appropriate choices, click **Add**.

Infolet Name	Job Role	OTBI Transactional Analysis Duty Role
- Agents with Most Pending Service Requests - Critical Service Requests - Long Wait on Agent - Queues by Unassigned Service Requests - Team Resolve Time	- Customer Service Manager - Sales Administrator	- Service Managerial Transaction Analysis Duty - Service Administrative Transaction Analysis Duty
- My Resolve Time - Pending Tasks - Waiting on Me - Waiting on Customer	- Customer Service Representative - Sales Administrator	- Service Transaction Analysis Duty - Service Administrative Transaction Analysis Duty

Related Topics

- [About Security Roles](#)

Analytics Subject Areas

Creating analyses begins with subject areas. A subject area is a functional grouping of the contextual entities (also called dimensions) by which the metrics (also called facts) can be analyzed.

Subject areas are the building blocks of analytic content in OTBI. For example, the columns in a tabular report showing the number of open (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 service application comes with a variety of standard subject areas for you to select from when you're building or editing your analytics.

.

Overview of Parent and Child Interactions in Analytics

With the support for cross-channel interactions, there is the prospect of an initiated interaction crossing channels and agents. Additionally, the ability to associate multiple business objects is also introduced as part of this functionality.

When a communication is initiated with a customer, an interaction is created that contains the channel, contact, resource, and multiple other attributes. Each interaction can have only one reference to each of these attributes. In the case of business objects, an interaction can be associated with one of each type of business object. In the case where a communication adds additional attributes, the application automatically generates a child interaction, and associates it to the original parent.

Here's a few examples of such cases:

- A communication crosses channels (the child interaction would store the new channel).
- A communication is transferred between agents (the child interaction would store the second agent).
- More than one type of business object is associated. For example, an interaction can be associated with one service request. However, if during the conversation, a second service request is created or updated, the second interaction would have to be stored in a child interaction.

For metric calculations in BI, it's often necessary to count the entire set of these atomic interactions, as a single interaction. This single interaction is referred to as the parent interaction. The atomic interactions are referred to as child interactions. The parent interaction count isn't the summation of child interaction count. In any cross-channel or cross-agent interaction, there's always a single parent interaction, and more than one child interactions.

Service Analytics Saved Queries

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.

Here's how you access the saved queries:

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**.

Here's a table that provides details about the available saved queries:

Analyses Name	Description	Job Role
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 - 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 - 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 - 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 - Sales Administrator
Queues where I am a resource	Saved query to retrieve the list of queues where the signed in user is a resource. Apply this on the Queue ID column when used in analyses.	- Customer Service Representative - 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 - Sales Administrator

Where do I find resources for Oracle Fusion Service Analytics?

Here's where you can find *additional information* on Fusion Sales, Service, and Help Desk analytics:

- *Prebuilt Fusion Service analytics* spreadsheet.
- *Creating and Administering Analytics* guide.
- *Subject Areas for Transactional Business Intelligence in Fusion Sales and Fusion Service* guide.

-
- *Security for Sales Analytics and Reports* guide.
 - *Securing Sales and Fusion Service* guide.
 - *Oracle Transactional Business Intelligence* guides.
 - *Fusion Service Report Sharing Center* cloud customer connect.
 - *Analytics and Reporting Best Practices* support resource center.
 - *R13.x CX OTBI Subject Area to Database Lineage Mapping Spreadsheet*
 - *About Creating Your Own Subject Areas*

31 Set Up Additional Service Features

Installed Base Assets

How do I enable Installed Base Assets for service requests and work orders?

If you use Installed Base Assets for processes such as Supply Chain, Service Logistics, Service Contracts, or IOT, you can opt-in to use the same asset model for SRs and Work Orders. Use Application Composer to add the Installed Base Asset fields to the SR and Work Order pages.

Enable Installed Base Assets

To opt in, do the following:

1. In the Setup and Maintenance work area, go to the following:
 - Offering: Service
 - Change Feature Opt in link
2. Click the **Features** icon for Service in the first row.
3. Select Enable for **Manage Assets Using Common Asset Model**.
4. Click **Done**.
5. Click Done on the Opt In page.

Note: This is a global setting where you choose whether to use Installed Base Assets or the default Asset object for the Service Request and Work Order process. You can't use both asset objects in Fusion Service so you should carefully consider the impact if you have requirements to support asset management outside these processes. For example, Installed Base Asset doesn't currently support sales processes in Fusion Sales and Service and has limited support for extensibility.

FAQs About Installed Base Assets

Frequently Asked Questions (FAQs) for Installed Base Assets

This section contains some frequently asked questions (FAQs) when setting up and maintaining installed base assets. Scroll or search to find answers to common questions.

Can I add Installed Base Asset as a subtab on the SR?

IB Asset is a single choice field on SR. It's not a subtab where you can associate many assets.

Where's the Installed Base Asset object in Application Composer?

The installed base asset object in Application Composer is called Installed Base Asset. It's located under the SCM/ERP drop-down list (not CRM).

How can I query Installed Base Asset-related fields in and SR from a particular object?

A few Installed Base asset fields are already part of the SR object.

There's a picker to choose which asset is the subject of the SR. It's possible to create formula fields and use groovy to access other fields from the IB Asset object.

Case Management

What's Case Management?

Case Management is a professional and collaborative process that provides the ability to manage long running processes to meet an organization's needs related to managing escalations, investigations, problems, applications, and other issues.

For example, a case is created whenever an application for specific services is requested by an applicant. Applications are typically assessed for eligibility of services, and upon successful assessment, a Case is created and managed. Unlike service requests, cases typically stretch over a long period of time, involve multiple parties as well as multiple documents and messages to deliver a solution or service, and often require complex business processes for successful completion.

The overall case management process is iterative, non-linear, and cyclical, with its phases being revisited as necessary, until the intended outcome is achieved. A Case can be different than a service request, as Cases are often employee, client, citizen, or student focused, and the outcome is often difficult to predict or measure. A case can be a process that ties together one or more services (benefits) delivered by an institution to a person (individual), group (household), or organization (customer account) to fulfill the specific intent and needs of the recipient.

For a detailed video presentations, see [Case Management for Employee Relations and Investigations \(HR Help Desk\)](#) on Cloud Customer Connect. (Sign in is required)

Opt Out of Case Management

Case Management comes ready to use with Fusion Service. If you don't wish to use Case Management, you can disable it.

To Disable the Case Feature

1. In the Setup and Maintenance work area, go to the following:
 - o Offering: Service

- Change Feature Opt in link
- 2. Click the **Features** icon for the Service offering.
- 3. Select Disable for **Case**.

Service Request Parts Ordering for Service Oracle Service Logistics Cloud Users

How can I give users access to create parts orders using service requests?

Service Logistics users access the Fusion Service UIs to order parts to complete a service request or work order. See related topic for instructions on granting users access to the Service Request and Work Order UIs so that they can create parts orders.

To view the table and column details which carry part details from the SR page, see the related link for the Parts Requirements Lines table.

Related Topics

- [Give Users Access to Fusion Service UIs](#)
- [Parts Requirement Lines](#)

Configure the Cancel and Delete Behavior for Service Requests with Parts Orders

If a service request is canceled or deleted before the parts on an order are sent, it's likely that the parts order should also be canceled to avoid unnecessary costs.

With the Manage Service Request Parts Order Disposition Options task, you can configure what happens when a user attempts to delete a service request that has parts on order. You can also specify which service request status values indicate the cancellation of a service request. The application prompts the user to confirm what they want to do before the status can be changed.

To configure the delete and cancel processing behavior:

In the Setup and Maintenance work area, go to the following:

- Offering: Service
- Functional Area: Service Request
- Task: Manage Service Request Parts Order Disposition Options

To configure the processing for if a user tries to delete an order with parts:

1. Click the **Execute part order delete processing** check box if you want the application to check to see if there are parts orders when the user tries to delete a service request.
If you do not check this box, a service request with parts orders is not affected if it's deleted. If you check this box, service requests that contain parts orders will be evaluated based on whether the next check box is checked.
2. Click the **Allow deletion of service request with noncancelable part orders** check box if you want to allow users to delete a service request even if there are parts orders that have already been sent to the customer.
If parts order delete processing is turned on, but this box is not checked, users can't delete a service request that has parts orders, unless all the parts orders are canceled. If this box is checked, users can delete the service request anyway. In this case, the application will also try to cancel the remaining parts orders that can still be canceled.

To configure the processing for when a user tries to cancel an order with parts:

Use the Service Request Resolved Status Type region of the page configure how the application responds when a user tries to set certain status values on a service request with parts orders that have already been sent to the customer.

Status values with a resolution type of Resolved suggest the assistance is no longer required. Some status values also might indicate that a problem was fixed. In cases where parts were ordered to help the customer fix the issue, but the status value suggests the customer might not need the parts anymore, you can configure the cancel processing to determine how the application responds.

1. Check the check box in the Cancel Part Order Process column for the status you want the application to run cancel processing. This means if the user tries to cancel an SR that's in that status, the application checks to see if there are any parts on the SR that have already been sent out. The user is then notified so they can decide if want to continue.

Copy Maps

How do I create a copy map?

Copy maps enable you to create objects by copying existing object entities. You use the copy maps feature to map child objects and attributes in the source object (From object) to objects and attributes in the object you're creating (To object).

To create a copy map:

1. Sign in as a user with Application Composer access and verify that you have an active sandbox.
2. Access Application Composer by selecting **Application Composer** from the **Navigator** menu, under the **Configuration** category.
3. Go to **Advanced Setup**, and select **Copy Maps**.
4. Under the Copy Service Request section, select one of the following options:
 - Copy Service Request Map
 - Standard Copy Service Request Map
5. If you select **Standard Copy Service Request Map**, a read-only page comes up where the attribute mapping, application module mapping, and entity mapping sections are already mapped.

6. If you select **Copy Service Request Map**, you can modify the attribute mapping, application module mapping, and entity mapping sections.
7. The **Attribute Mapping** section contains all the OOTB attributes that are copied automatically. In this map, you can add the following types of fields to be copied:
 - o other standard/OOTB fields,
 - o custom fields, and
 - o fields from other objects.

In the **Application Module Mappings** section, enter the field you want to copy. You enter the following information about the mapping as outlined in this table:

Field	Description
Name	Enter a name for the copy map. The name is displayed in the Copy Maps window. You can use the name to locate and search for copy maps that you want to edit or delete.
Map Group	Select the required mapping group for the new copy map from the drop-down list.
To	Select the application module that you're copying to. For example, if you're copying leads to opportunities, then select the application module that includes leads.
From	Select the attribute or field name that you're copying from. For example, if you're copying responses to leads, then select the application module that includes responses.

8. In the **Entity Mappings** section, add a record for each view object that you're copying by completing these fields:

Field	Description
To	Select the new object that you're copying to.
From	Select the existing view object that you're copying from.
Query Type	Select one of the three query types: Unique Identifier, Join, or All Records. If you select All Records, then don't enter any information in the remaining fields.
Query Attribute	If you selected either Unique Identifier or Join in the Query Type field, then select an attribute to use during runtime. The attribute is used to filter the records that are copied to the new object. For example, if you select LeadId, then during runtime, only the records matching the Lead ID passed to the copy map engine are copied.

Field	Description
Joined View Object and Joined Attribute	If you selected Join in the Query Type field, then use these two fields to specify: ○ The view object to which this object is joined ○ The attribute used as the basis of the join

9. Select each line in the **Entity Mappings** section, and add or update records to the **Attribute Mappings** section for each of the attributes by completing these fields:

Field	Description
To	Select the attribute to which the selected entity is copied.
Referenced View Object	If this attribute is a foreign key, select the view object joined by this foreign key. The application generates new foreign keys that keep the reference intact.
Primary Key	If this attribute is a primary key, select this check box. Instead of copying the value in the From object, the application generates a unique value for this key field in each record in the To object.
From	Select the attribute from which the selected entity is copied.
From Expression	Optionally, you can enter a Groovy expression to change the value in this attribute. For example, you want to change the value of the From object to some new value in the attribute of the To object. You can also enter a constant to fill this attribute with a constant value in every record of the To object.

10. Click **Save and Close**.

32 Integrate Oracle Internet of Things Service Monitoring for Connected Assets

Overview of Oracle Internet of Things Service Monitoring for Connected Assets

Oracle Fusion Service and Oracle Internet of Things Service Monitoring for Connected Assets integration is designed to support customers who want to use the 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 the service application 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 a service request.
- Closing the Internet of Things Service Monitoring for Connected Assets incident when the corresponding service request is resolved.
- Viewing information about the connected asset directly from the service request pages in the service application. 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 [Related Topics](#).

Related Topics

Integrate Engagement Cloud with Internet of Things Service Monitoring for Connected Assets

How You Integrate the Applications

Here's a brief overview of the steps required to integrate Fusion Service with Internet of Things Service Monitoring for Connected Assets.

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 Fusion Service and Internet of Things Service Monitoring for Connected Assets, if the instances use self-signed certificates.
4. Activate connections to Fusion Service 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 [Related Topics](#).

Related Topics

Create an Integration User

To integrate Fusion Service 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 Fusion Service service catalog or event catalog web services from Oracle Integration Cloud.

Here's how you create an integration user and the privileges you must provide to the role:

1. Sign in to the service application as an administrator.
2. Select **Navigator > My Team > Users and Roles**. The Search Person page is displayed.
3. Click **Create**. The Create User page is displayed.
4. Enter the following details for the new user:
 - **Last Name**: Enter a last name for the user.
 - **Email**: 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 that 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 can't 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

Related Topics

- [Administering Oracle Integration: Add Users, Groups, and Roles for an Existing Instance](#)

Import the Integration Package

You can set up the Integration Cloud integration package after you set up Fusion Service and create the integration user with the required privileges.

Before you start setting up Integration Cloud, go to the Oracle Marketplace (https://cloudmarketplace.oracle.com/marketplace/en_US/homePage) and search and download the Fusion Service to Internet of Things Service Monitoring for Connected Assets integration package. For more information about Oracle Marketplace, see the Related Topics link for Oracle Cloud Marketplace documentation.

The Integration Cloud integration package supports the following:

- Inbound: Create an SR in Fusion Service 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.

To import SSL certificates for Fusion Service, see "Uploading an SSL Certificate" in the *Using the Oracle Sales Cloud Adapter* guide, available at <https://docs.oracle.com/en/cloud/paas/integration-cloud-service/icssc/index.html>.

To import SSL certificates for Internet of Things Service Monitoring for Connected Assets, see "Uploading an SSL Certificate" in the *Using the REST Adapter* guide, available at <https://docs.oracle.com/en/cloud/paas/integration-cloud-service/icsre/index.html>.

Activate the Connections

After you download the package and import the SSL certificates, connect to Fusion Service using the sales 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>.

Configure the Connection to the Fusion Service Instance

Here's how you configure the connections to your Fusion Service 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 Fusion Service is listed on the connections page.
4. Click **Oracle Fusion Service** 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 Fusion Service instance.
 - **(Optional) OSC Event Catalog URL:** Enter the event catalog URL on your Fusion Service 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 service application page.
10. Ensure that the connection test is successful and the status meter shows 100%.
11. Click **OK**.

Configure the Connection to the Internet of Things Service Monitoring for Connected Assets instance

Here's how you 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
5. Configure Security.
6. Save the configuration and test the connection.

Activate the Integrations

After you configure your connections, you must then activate these integrations:

- **Autocreate SR for IoT Incident:** Creates or updates an SR in Fusion Service 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 Fusion Service.

Here's how you activate the integrations:

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 that the flow has been activated successfully.
7. Repeat the previous steps for each integration.

Enable the Connected Asset Tab

Here's how you enable the Connected Asset tab on the user interface.

1. Import the SSL certificate from the Internet of Things Service Monitoring for Connected Assets instance to the Fusion Service server.
2. Sign in to the service application 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.
7. (Optional) To configure the roles and users who can perform the actions in the Actions and Attributes section of the Connected Asset tab of the Edit Service Request page, do the following:
 - a. In the **Service Roles to IoT Users Mapping** section, click **Create**.

A blank row is added in the table, which enables you to map the Service roles to the Internet of Things application users.
 - b. Enter the **Role**, **Priority**, **User Name**, and **Password**.

If a user has multiple roles assigned, then the role with the highest priority is considered. Lowest number indicates highest priority.
 - c. Click **Verify Connection** to verify the connection.
 - d. Click **Save**.

On the **Connected Asset** tab, the actions are displayed in the **Asset Actions** drop-down list based on the signed in user's role that's used to connect to the Internet of Things Service Monitoring for Connected Assets instance.

Import Internet of Things Connected Assets

You can import Internet of Things connected assets from an external data source into Fusion Service using the Import and Export Management feature.

