Oracle Fusion Service

Implementing Digital Customer Service

Oracle Fusion Service
Implementing Digital Customer Service

F77882-12

Author: David Yetter

Contents

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

This guide outlines the implementation and configuration steps required to develop, configure, manage, and administer Digital Customer Service in Oracle Fusion Service. To set up and work with the additional features of Oracle Fusion Service, see Oracle Fusion Sales and Fusion Service documentation on

This implementation guide is designed to be used as a starting point that shows how Digital Customer Service in Oracle Fusion Service can be developed and implemented using Oracle Visual Builder. An implementor can use the documented development and configuration information in this guide to successfully deploy Digital Customer Service.

Each implementation of Oracle Fusion Service is unique, and leads to the development of customer-specific applications that support their unique business requirements.

This guide describes how to deploy and configure Oracle Digital Customer Service in conjunction with Oracle Visual Builder **Visual Applications**. If you're using Oracle Visual Builder **Classic Applications** for your Digital Customer Service application (even if your Oracle version is 18B or later), refer to Implementing Digital Customer Service in Engagement Cloud Release 13 (Update 18B). The Implementing Digital Customer Service in Engagement Cloud Release 13 (Update 18B) is the documentation that's pertinent to the **Classic Applications** development model.

> **Note:** This document describes features available to users under Oracle Fusion Sales, Oracle Fusion Service, and Oracle Engagement Cloud licensing agreements.

Related Guides

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

Title	Description
Oracle Cloud Developing Applications with Oracle Visual Builder	Describes how to use a web-based visual development tool to create and publish custom web and mobile applications that can integrate business objects and applications REST services to extend SaaS services.
Oracle Cloud Administering Oracle Autonomous Visual Builder Cloud Service	Describes tasks for administrators of Oracle Visual Builder.
Using Service	Contains information to help service managers, service personnel, and other service end users to perform day-to-day business tasks using Oracle Cloud.
Using Knowledge	Describes how administrators, agents, authors, and other knowledge base contributors can implement and use Knowledge.

Title	Description
Implementing Service	Describes how to set up Service Request Management.
Getting Started with Your Sales Implementation	Describes your initial implementation procedures, based on a simple sales-force-automation use case.
Implementing Sales	Contains conceptual information and procedures needed to implement components and features.
Securing Sales and Service	Contains information to help setup users and sales administrators configure access to functionality and data.
Security Reference	Lists the predefined security data that's included in the offering.

2 About Digital Customer Service

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.

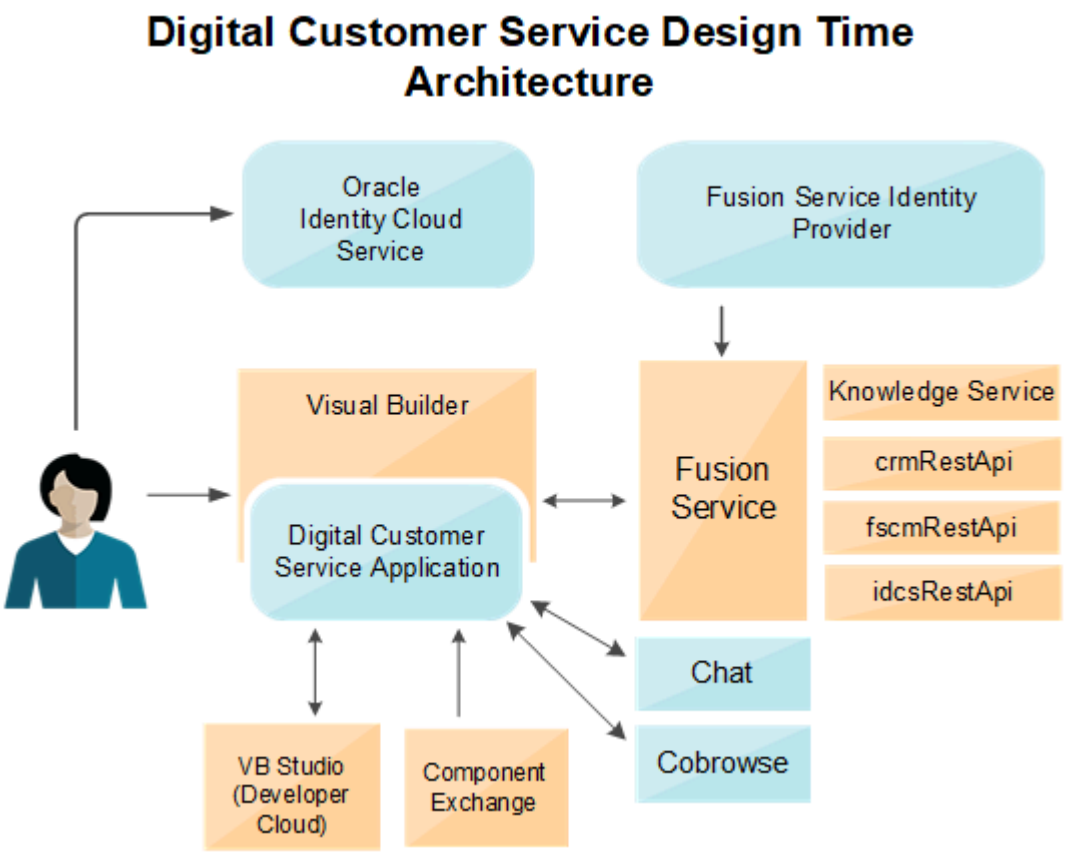
> **Note:** To use Chat inlays in Oracle Fusion Service, you must configure some profile options. For more information, see "Configure Chat Inlay" in *Oracle Fusion Service Implementing Digital Customer Service*.

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.

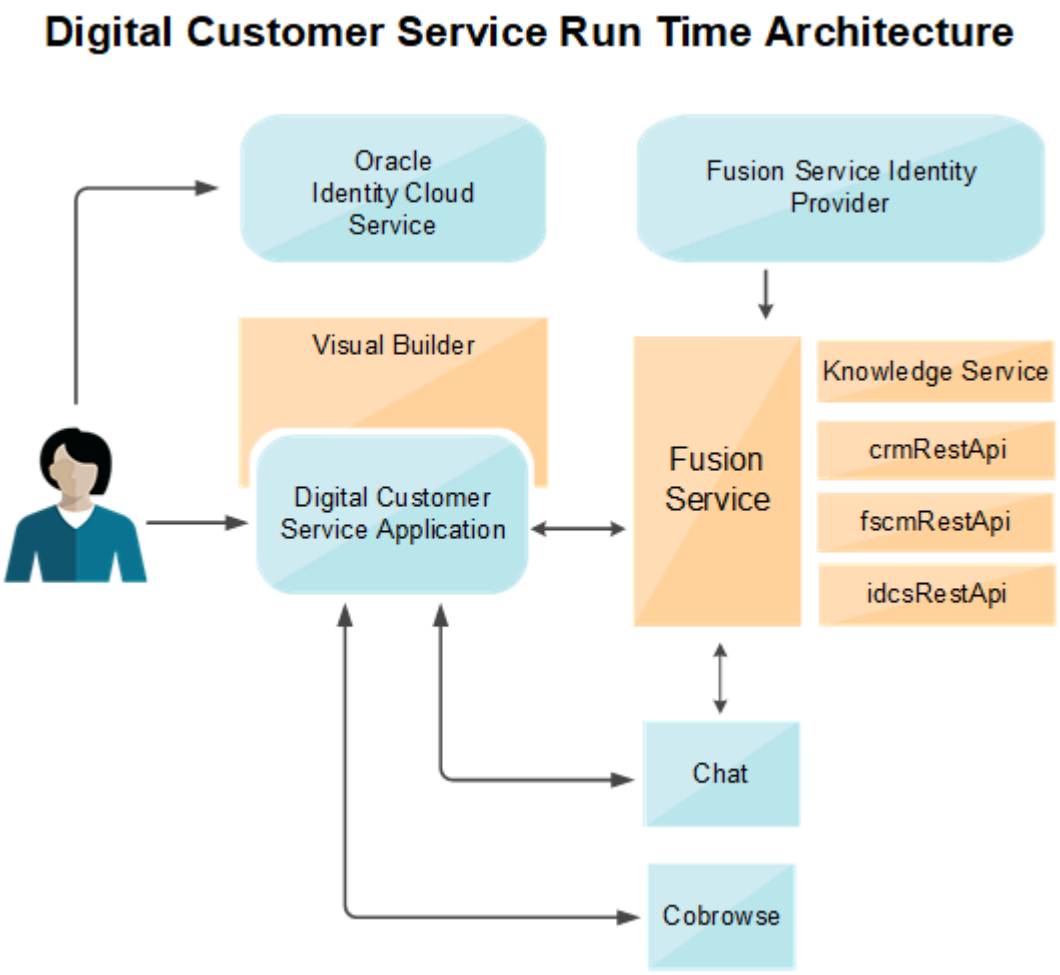
Here are general overviews of the design time and of the runtime architecture. The Digital Customer Service application relies on the following APIs:

- **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:



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 Terminology

Here's an overview of some Digital Customer Service terminology.

- **Oracle Visual Builder**: A cloud-based visual development tool that gives you easy access to data from any REST-based service. You can create and test responsive web applications and native mobile applications without installing any additional software. The visual designer lets you quickly lay out pages in your applications by dragging and dropping UI components, configuring their attributes and defining their behavior.
- **Root Page**: A page that contains the shell of your application including the header and footer and navigation components.
- **Service APIs**: The REST APIs that your application interacts with.
- **Digital Customer Service Application**: The application that you build using Oracle Visual Builder that acts as the web interface for your users. The Digital Customer Service application is built using the Digital Customer Service Reference Implementation template.
- **Digital Customer Service Users**: Your customer account users who have successfully registered to use the Digital Customer Service application as self-service users. These users can have a variety of roles.

- **Digital Customer Service Templates**: The available templates you can select while creating your Digital Customer Service application. These application templates include component extensions, themes, and predefined pages and actions.
- **Digital Customer Service Reference Implementation**: This template includes several pages and business components that enable a basic support experience including: knowledge search, service request creation and management, display of work orders, chat, and self-service user management capabilities for the account administrator. The account administrator manages all of the users and roles.

For more information on roles, refer to *Digital Customer Service Roles*.

Related Topics

- Developing Applications with Oracle Visual Builder

What are the Digital Customer Service roles?

You can grant different roles to Digital Customer Service users in Fusion Service and Oracle Identity Cloud Service. Here are some details about the privileges granted with each role in Fusion Service.

User

The User application role is mapped to the Customer Self-Service User job role in Identity Management. The role grants the privileges to view and edit service requests created by the user, and to create service requests. The role also grants the privileges to view and update work orders on which the user is the contact. Removing this role causes the removal of all privileges. The only way to restore the privileges is to submit a new registration request. The role also grants the privileges to view interactions on which the user is the primary contact.

Account Manager

The Account Manager role grants the privileges to view and edit all service requests and view and edit all work orders for a specific customer account. Additionally, users with the Account Manager role can create service requests. The user can perform these tasks only on accounts that they're the Account Manager for.

Account Administrator

The Account Administrator application role is mapped to the Customer Self-Service Account Administrator job role in Identity Management. The role grants the privileges to view and approve registration requests in the customer account for which they're the Account Administrator. Additionally, the role permits the user to assign and remove the Account Administrator and Account Manager roles.

Related Topics

- How do I manage Self-Service users?

Overview of Self-Service Users

In the past, Fusion Service used its own identity provider for authentication and authorization.

As a result, Digital Customer Service self-service users who had to access data in Fusion Service were required to have an account in both Oracle Identity Cloud Service and in the Fusion Service identity store. The self-service registration process had to create user accounts in the Fusion Service identity store, and then synchronize them to Identity Cloud Service. You also couldn't configure the Fusion Service login page to match your corporate brand requirements.

Along with these limitations, the Fusion Service identity provider is limited in its ability to scale as it was designed more for Fusion applications rather than B2C applications. Many of the implementations oriented more to B2C require more scalability.

Using this implementation, users can access data in Fusion Service using accounts that reside solely in Identity Cloud Service. When a user submits a self registration request, it goes to Fusion for approval. Once approved, the user account is created in Identity Cloud Service and a simple contact record is created in Fusion Service. In addition, the Identity Cloud Service sends out a welcome email to the user at which point the user can create their own password.

The contact record has the IDCS user GUID which provides a link to the IDCS user. When a REST API request is made the user is mapped to a set of proxy users that are predefined in Fusion Service. Each of the proxy users can execute a set of predefined APIs.

When the user logs in to the application the only login option is by way of Identity Cloud Service. The log in information can be configured by the user. All management of users is done using the Identity Cloud Service administrations console. There's no need for user synchronization for self service users as they're already present in Identity Cloud Service.

Benefits of Self Service Optimization

Self-Service Optimization gives you the following benefits:

- A log in page that you can configure to match your corporate identity.
- Fusion Service is still the identity provider for Fusion Service users.
- A local identity provider created in IDCS for Digital Customer Service self-service users.
- Self-service users log in using Digital Customer Service through IDCS.
- Self-service users are created in IDCS rather than in Fusion Service.
- There's no need for user sync from Fusion Service to IDCS, thereby allowing users to log in immediately after self-service registration approval.
- User accounts no longer need to be created in Fusion Service.
- There are only a small number of proxy users representing all self-service users in Fusion Service.
- Self-service users are still represented as customer contacts in Fusion Service.

The Proxy User

You define proxy users in Fusion Service. In general, there should be one proxy user per self-service persona in Fusion Service. For example, the Digital Customer Service application can be used by either Customer Self-Service users or Customer Self-Service Account Administrators.

Since there are two personas, two proxy user accounts must be created and configured. You must also create an anonymous proxy user. The two proxy users that correspond to Digital Customer Service application roles are:

- Customer Self-Service Users.
- Customer Self-Service Account Administrators

You give the proxy user all the functional privileges or roles required by the persona. If you create a proxy user account for the Customer Self-Service Users persona give that account the Customer Self-Service User role. When you create a proxy user account for the Customer Self-Service Account Administrators persona give the account the Customer Self-Service User and Customer Self-Service Account Administrator roles.

A proxy user account is used by multiple actual users to perform REST operations in Fusion Service. An actual user accessing data gets the functional privileges from the proxy user but the data privileges are based on the GUID of the actual user in Identity Cloud Service. This ensures that a given user will be able to see relevant data.

Functional privileges are those that relate to actions a user can perform in user interface pages, whereas data privileges are those which concern which data a user can access.

Self-Service Optimization Architecture

Self-Service Optimization gives you greater flexibility by using proxy users in Fusion Service.

Whereas both a contact and a user were formerly created in Fusion Service, now only a contact record is created in Fusion Service, and users, created in IDCS, map to proxy users that are few in number and created during initial setup.

The proxy users define the functional privileges that are accessible to the users. A distinct proxy user represents each persona using the application.

Self-Service Optimization Security

When the user is created in Identity Cloud Service a GUID attribute is created. This attribute maps to a GUID field in the Service Contact record.

Here's how authentication works:

- Identity Cloud Service is the identity provider for self-service users. Identity Cloud Service uses a local identity provider to authenticate the self-service users.

- Self-service user accounts are only created in Identity Cloud Service. You just need to set the SVC_CSS_USE_FA_AS_IDP profile option to **False** to control this behavior. For more information, see the link to the Set Profile Options topic in Related Topics.
- You have one proxy user for each application persona. The Customer Self Service User and the Customer Service Administrator are two personas, and two separate roles.
- Each self-service user is associated with a customer contact record in Fusion Service. The GUID field in the Identity Cloud Service user record maps to the GUID field in the Fusion Service contact record.
- Data in Fusion Service is accessed using a proxy user account with appropriate functional privileges. Proxy user data service manages access to data in Fusion Service with the appropriate proxy user.

Functional Privileges of the Proxy User

Here's an overview of the functional privileges of the proxy user.

- Has all the functional privileges given to the proxy user role.
- Has the FND_IDP_PROXY_USER_PRIV privilege allowing it to act as a proxy user.

Data Privileges of the Proxy User

Data privileges given to the proxy user vary dynamically based on the actual user session. The data privilege of the proxy user is determined by using the proxy user authentication mechanism:

- The GUID of the authenticated user is taken from the HTTP header and stored into the session in Fusion Service.
- The GUID is used to look up the PARTY_ID of the appropriate contact.
- Data security policy predicates are based on the PARTY_ID of the contact.

Related Topics

- How do I set profile options for Self-Service Optimization?

Technical Compatibility

This topic describes the software versions that are compatible with this release of the Oracle Digital Customer Service offering.

Oracle Digital Customer Service is a cloud service. Oracle Digital Customer Service is delivered as a reference implementation application template on Oracle Visual Builder. When you create a new application for Oracle Digital Customer Service, use the template that matches the version of your Oracle Fusion Service. So, if you're using the latest release, use the template that matches that release number, and if you're using an older release, use the template that matches that release number.

- The Oracle Identity Cloud Service license provided with Digital Customer Service is Oracle Identity Cloud Service Foundation. For more details, see the Related Topics section.
- Oracle JavaScript Extension Toolkit (Oracle JET). Oracle Visual Builder includes Oracle JET bundled in with it and updates the version to uptake new features periodically. Oracle Digital Customer Service is compatible with the Oracle JET versions bundled with the corresponding versions of Visual Builder.

- Browsers supported by Oracle Visual Builder are listed in the Supported Browser topic in the Oracle Cloud Known Issues for Oracle Visual Builder. See the Related Topics that follow.
- Oracle Digital Customer Service isn't Break-Glass compliant as it depends on Oracle Identity Cloud Service.
- Digital Customer Service isn't supported with Symantec Blue Coat Cloud Data Protection Gateway.

How to Determine the Version of Oracle Fusion Service

To determine the version of Fusion Service that you have installed:

1. Sign in to Fusion Service.
2. Select the menu next to the name of the signed-in user.

   The **Settings and Actions** menu appears.
3. Click **About This Application**.

   The **About This Application** dialog box is displayed. The version number appears after the word **Revision**.

Related Topics

- Determine Component Versions
- About Oracle Identity Cloud Service Pricing Models
- REST API for CX Sales and Fusion Service
- Known Issues for Oracle Visual Builder

3 Mandatory Setup Tasks

Overview of Required Setup Tasks

Here are the required tasks that are covered sequentially in this chapter. You can use these steps, like a checklist, to get your Digital Customer Service application set up:

> **Note:** If you're using Self-Service Optimization, you must choose Oracle Identity Cloud Service as your primary identity provider. Self-Service Optimization uses proxy users in Fusion Service which streamlines setup and provides you with greater flexibility. In this architecture, only a contact record is created in Fusion Service, and users, created in IDCS, map to proxy users that are fewer in number and created during initial setup. For more information, see the Overview of Self-Service Optimization topic in this chapter.

1. Activate Digital Customer Service
2. Set Up Oracle Identity Cloud Service
    a. Find the Oracle Identity Cloud Service URLs
    b. Set Up Oracle Identity Cloud Service for Authentication
    c. Create the Application Client
    d. Create Groups in Oracle Identity Cloud Service
    e. Configure the Resend Welcome and Password Recovery Email Templates
3. Set Up Fusion Service
    a. Enable Digital Customer Service
    b. Configure a User Category for Proxy Users
    c. Create the Proxy Users
    d. Set Profile Options for Self-Service Optimization
    e. Set the Oracle Identity Cloud Service Endpoint
    f. Configure the Client ID and Client Secret
    g. Manage the Proxy User Configuration Data
    h. Set Proxy User Credentials
4. Set Up Administrators and Developers
    a. Overview of Digital Customer Service Developer Roles
    b. Create an Internal Customer Account
    c. Create Digital Customer Service Developers
    d. Add Visual Builder Roles
5. Set Up Oracle Visual Builder
    a. Retrieve the Oracle Visual Builder Designer URL
    b. Verify Your Oracle Visual Builder Settings

Considerations for Setting Up Cloud Accounts

Most Oracle Cloud services are designed to run in an Oracle Cloud account paired with Oracle Identity Cloud Service (IDCS). However there's also what's known as a Traditional Cloud Account which uses traditional identity and access management software.

Traditional Cloud Accounts have a 1:1 mapping between the Identity instance and the Traditional Cloud Account.

An Oracle Cloud Account uses IDCS to manage users and control access to cloud services. IDCS allows a one-to-many mapping, which means many instances can be associated with a single cloud account.

To learn more about cloud accounts, see the Applications Services link in the Related Topics section.

> **Note:** It's strongly recommended that you choose the same Cloud account as the one that your implementation of Fusion Service resides in.

When setting up your cloud account note that the name you choose will also be used to create the URLs to access all your cloud services and once you create the account name you can't change it, so take extra care in creating your cloud account name.

As an example, say you call your Oracle account "abccorp." The URL for your cloud account might look something like this:

```
https://<fusion applications family>-odcsvbcs-05281907-1391-<abccorp>.builder.ocp.oraclecloud.com/ic/
builder.
```

Also, the cloud account name must have the following:

- Must be unique.
- Must start with a lowercase letter.
- Can have up to 25 lowercase letters and numbers.

Related Topics

- Applications Services

How do I activate Digital Customer Service as a new customer?

To perform this step you must have the Service Administrator role. The Service Administrator receives the "Action Required" welcome email from Oracle. As the designated activator of the Oracle Digital Customer Service service, the activator is only required to activate Digital Customer Service.

After that, the activator can select a different service administrator to manage the day-to-day administration of the service during the activation process if necessary.

1. Open the email prefaced "Action Required" that you received from Oracle Cloud.

2. Review the information about your service in the email, and then click the provided link to activate your service.
3. In the Activate My Service form, do the following:
    a. Enter a cloud account name.
    This name is used to identify your cloud account. We recommend that you use the same Oracle Cloud account that Fusion Service resides in.
    b. Enter Administrator details, and if you're not going to be the Service Administrator going forward, assign the new Service Administrator now.
4. Click **Create Account** to proceed to submit your request.
5. Click **Close**.
    The account is now active and ready to use.

How do I activate Digital Customer Service as an existing user?

To perform this step you must have the Service Administrator role. The Service Administrator receives the "Action Required" welcome email from Oracle. As the designated activator of the Oracle Digital Customer Service service, the activator is only required to kick off the provisioning process.

1. Open the email prefaced "Action Required" that you received from Oracle Cloud.
2. Review the information about your service in the email, and then click the provided link to activate your service..
3. Enter your cloud account name, and click **Next**.
4. Click **Continue** on the Cloud Tenant page.
5. On the log in page, enter your cloud account credentials, and click **Sign In**.
    The My Services page appears.
6. Click the **Manage Account** tile.
7. On the Account page, click the **Activate** tab.
8. Choose the service you want to activate, and click the **Cloud Services Account Setup** button.
9. Click the **Cloud Account Name** drop down list and select the cloud account you want to activate the service into and then click the **Assign Account** button.
    You receive a Review Summary message when the order is successfully activated.
10. Click **Close** to complete the activation phase.
    The account is now ready to use.

Set Up Oracle Identity Cloud Service

Required Role for Identity Cloud Service Setup

To perform setup in Identity Cloud Service, you must have the **Identity Domain Administrator** role.

Locate the Oracle Identity Cloud Service URLs

After activating your new Oracle Cloud account, here's how you locate your Identity Cloud Service URLs.

1. Log in to your Oracle Cloud account by navigating your browser to: https://www.oracle.com. This account is also known as your Administrator Account.
2. Click **View Accounts**, then **Sign in to Cloud**.
3. Enter the Cloud Account name if known, otherwise click **Forgot Your Cloud Account Name? Get Help**.

   > **Note:** The Cloud Account often corresponds to the name of the production pod

4. Now, click the Identity Cloud tile on the My Services dashboard.

   In your Cloud Account there's an **Identity** instance along with additional instances corresponding to and named after each Fusion Service instance. There's a 1:1 mapping between IDCS instance and each Fusion Service instance.
5. Click the tile or link for the Identity Service Instance that corresponds to the pod name and log in using your Oracle Identity Cloud Service Administrator credentials.

Set Up Oracle Identity Cloud Service for authenticating your VBCS Application

By default the security policy defined is set up to use Fusion for sign in. For DCS applications created with VB, the policy should be changed to have IDCS provide the sign in. This is true for when using the DCS Custom Sign In components.

After this change, the new policy defined for VB will be for all VB applications. Optionally you can configure Identity Cloud Service. If there are VB applications embedded in Fusion, these should be explicitly listed in a new policy or added to the default policy.

This means two policies are created, one for IDCS sign in (user name-password) and one for Fusion sign in (Oracle Fusion Application). The non embedded Visual Applications should be selected to be used for IDCS sign in and the embedded Visual Applications should be selected for Fusion sign in respectively.

1. In your web browser, enter: `https://cloud.oracle.com` and sign in to your Oracle Cloud account.

   > **Note:** This step is optional, and only required on your first use.

2. Sign in to Identity Cloud Service as the Administrator user.

   > **Note:** Navigate to the domain first if required.

3. Click the Navigation drawer and then expand the **Security** list.
4. Click **IDP Policies**, and then click **Add**.
5. In the **Policy Name** field, enter a policy name, such as VBCS Sign in Policy, and click **Next**.
6. On the Add Identity Provider Roles page, click the **Add Rules** button.
7. In the Rule Name field, enter a rule name, then click the **Assign Identity Provider** field and choose an identity provider.
8. Click **Save**, and then click **Next**.
9. On the Apps page, click the **Assign Apps** button to add the VBCS application.
10. Select the application starting with VBINST_ or `<Fusion family name>*_vb_` and click **OK**.

11. Click **Finish** to complete and save the settings.

Create a Client application

A Confidential Client application must be created for use with Fusion Service when it interacts with IDCS. Use the IDCS Administrator Console to obtain appropriate application client credentials.

You'll either create an application client or get the Client ID and Client Secret of an existing client application.

If you need to create an application to use, follow these steps:

1. Sign in to Identity Cloud Service as the Administrator user.
2. Click the Navigation drawer and then expand the **Applications** list.
3. Click the **Add** icon.
4. Choose **Confidential Application**.
5. In the Name field enter **Proxy User Data Service App** and click **Next**.
6. Select **Configure this application as a client now** option.
7. In the Authorization area, select the **Client Credentials** option for Allowed Grant Types.
8. In the Token Issuance Policy area, for **Grant the client access to Identity Cloud Service Administrator APIs**, click **Add** or select all.
9. In the **Add App Role** dialog box, select **User Administrator**, and click **Add**.
10. Click **Next**, and then on the Resources page, click **Next** again.
11. In the **Web Tier Policy** page, click **Next** again.
12. On the **Authorization** page, click **Finish**.
13. From the **Application Added** dialog box, record the **Client ID** and **Client Secret** and then click **Close**.
14. Click **Activate**, and then in the confirmation dialog box, click **OK** to activate the application.

How do I configure the Resend Welcome and Password Recovery email templates?

As part of user registrations, a user is created in IDCS and initially the user record is inactive.

After all the data related to the user is stored in the appropriate locations, the user is activated again and an API call is made to resend the Welcome email. To prevent a user from receiving multiple emails and to ensure the user sees the appropriate welcome message you've two configuration tasks depending on the scenario.

- You use the Resend Welcome template when a new user has signed up.
- You use the Password Recovery template when the user has been migrated.

Here are both tasks:

Configure the Resend Welcome Template

1. Sign in to Identity Cloud Service as a user with Administrative access
2. From the Navigation drawer, select **Settings**, and then **Notifications**.
3. Click the **Configure** tab.
4. Deselect the **Welcome** and **User Activation** check boxes.
5. Make sure the **Resend Welcome** checkbox is selected.

6. Click **Save**, then when the confirmation dialog box appears click **OK**.
7. Click the **Email Templates** tab.
8. Expand the **Resend Welcome** template, and replace the existing Subject line and message text with you own information and click **Save**.

Configure the Password Recovery Template

1. Sign in to Identity Cloud Service as a user with Administrative access
2. From the Navigation drawer, select **Settings**, and then **Notifications**.
3. Click the **Configure** tab.
4. Deselect the **Welcome** and **User Activation** check boxes.
5. Make sure the **Password Recovery Request** checkbox is selected.
6. Click **Save**, then when the confirmation dialog box appears click **OK**.
7. Click the **Email Templates** tab.
8. Expand the **Password Recovery Request** template, and replace the existing Subject line and message text with you own information and click **Save**.

Set up Self-Service registration when IDCS is configured as a service provider

You can configure Identity Cloud Service to act as a service provider to other external identity providers.

In this scenario the self-service registrations made when the value of the profile option SVC_CSS_USE_FA_AS_IDP is False will continue to create user accounts only in IDCS.

Depending on the identity provider, a custom process must be used to synchronize the user account with the external identity provider.

As the identity provider in this scenario isn't IDCS, the external provider will now have the responsibility for sending out the welcome email, managing the password cycle, and performing authentication. Also, IDCS must be configured to not send out the welcome email that allows a user to set the password.

Here's how you configure IDCS to not send the welcome email;

1. Sign in to Identity Cloud Service as a user with Administrative access.
2. From the Navigation drawer, select **Settings**, and then **Notifications**.
3. Click the **Configure** tab.
4. In the **End-User Notifications** work area, deselect the following check boxes:
    - Welcome
    - User Activation
    - Resend Welcome
5. Click **Save**, and when the confirmation dialog box appears click **OK**.

How do I add Visual Builder roles?

You must add roles in Oracle Visual Builder and Oracle Visual Builder Studio to configure appropriate authorization for your Oracle Visual Builder platform and editor.

Instead of assigning roles to each user individually, you can simplify the process by creating groups and adding users to groups as a way to assign roles. Each time you add a user to a group, the user automatically gets the roles defined for the group. Or you can assign individual users to the Visual Builder roles.

Add Users to a Group

Add users to a group so that they automatically get the permissions defined for the group.

1. Sign in to Oracle Identity Cloud Service Console as an administrator.
2. From the Oracle Identity Cloud Service console, click the **Navigation menu** icon.
3. Click the **Groups** tab.
4. On the Groups page, click the group that you want to assign users to or add a new group.
5. On the Group Details page, select the **Users** tab.
6. Click **+ Assign**.
7. Select the users you want to add to the group, then click **OK**.
   The selected users are added to the group. You can now assign roles to all the users in the group, if required.

Assign Users or Groups to Roles in Visual Builder

After you create groups and add users to groups, you can assign roles and provide access to services and instances to all the members of the group at once.

Assign Roles to Users in Visual Builder

1. In Identity Cloud Service, click the **Navigation Drawer**, and select **Oracle Cloud Services**.
2. Select your Visual Builder instance.
   You can use the filter to help you find your instance. For Oracle Visual Builder Studio instances search for `visualbuilder`.
3. Click the **Application Roles** tab.
4. Click the menu options icon shown next to the role, and select **Assign Users**. To assign the role to a group, you need to select **Assign Groups**.
5. Assign the Oracle Visual Builder **Service Developer** role to the group or users that require Build and Maintain access in the Oracle Digital Customer Service applications.
6. Assign the Oracle Visual Builder **Service Administrator** role to the group or users that require Administrative access to configure instance-wide settings for the Oracle Visual Builder environment for all applications.

Assign Roles to Users in Visual Builder Studio

> **Note:** The following VB Studio steps apply only to one pod, typically your TEST instance. VBCS, however, is provisioned on all pods so those steps for the application roles are applicable to all instances.

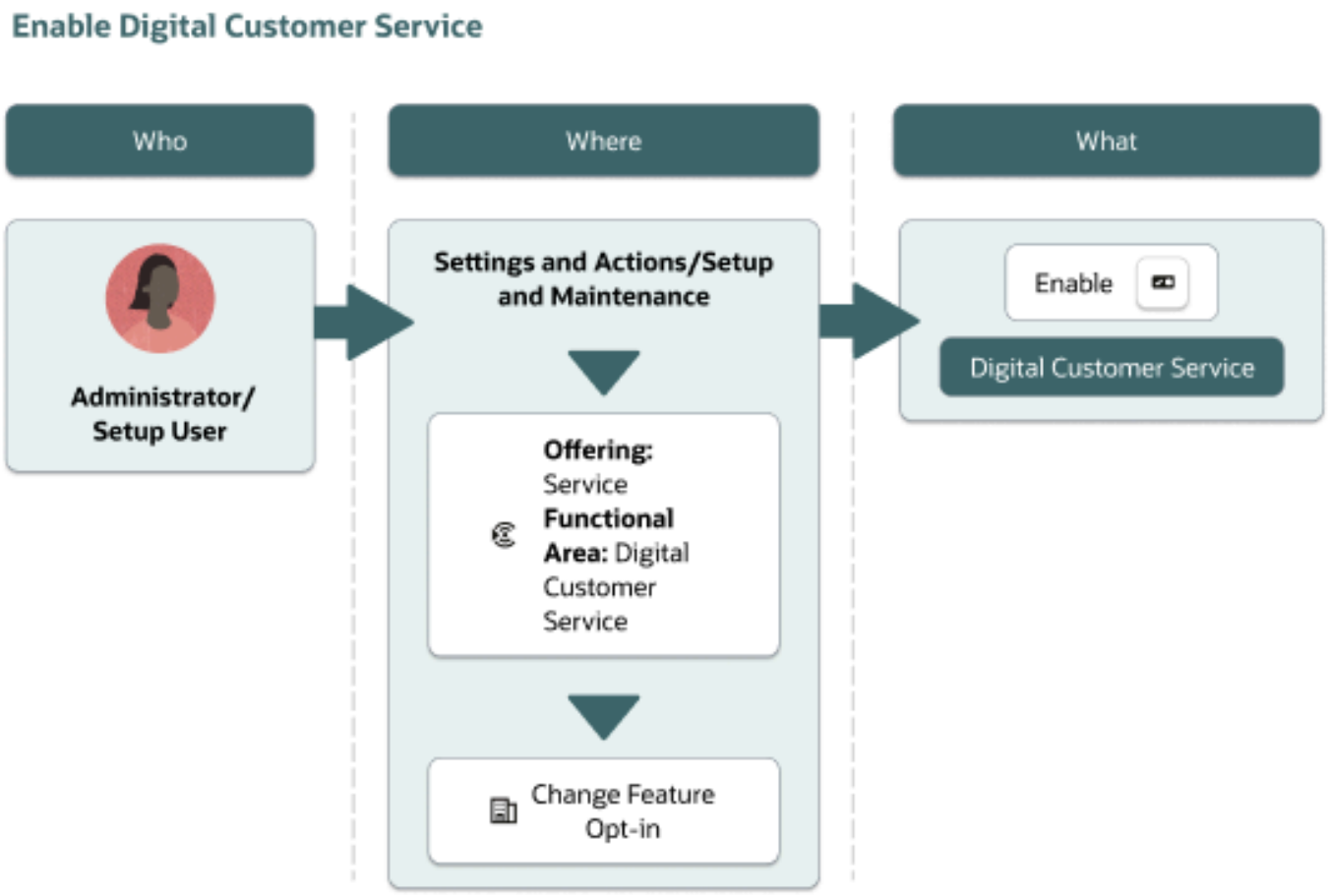
1. Sign in to Identity Cloud Service, click the **Navigation Drawer**, and select **Oracle Cloud Services**.
2. From the Oracle Cloud console dashboard, navigate to the Identity Cloud console and click **Applications**.
3. Click the link for your Oracle Visual Builder Studio instance.
   You can use the filter to help you find your instance. For Oracle Visual Builder Studio instances search for your instance prefixed with "VBINST".
4. Click the **Application Roles** tab.
5. Click the menu options icon shown next to the role, and select **Assign Users**. To assign the role to a group, you need to select **Assign Groups**.
6. Select the checkbox next to the name of each user that you want to add to the role, and then click **Assign**.

Note: You must add the DEVELOPER_USER or the DEVELOPER_ADMINISTRATOR user roles.

Set Up Oracle Fusion Service

Enable Digital Customer Service

You must enable Digital Customer Service in Fusion Service to have access to the profile options and work areas.



Here are the steps for enabling the Digital Customer Service application:

1. Sign in to Fusion Service as an administrator or a setup user.
2. Click the **Settings and Actions** drop-down list, and select **Setup and Maintenance**.
3. From the **Actions** drop-down list, select **Go to Offerings**, and then select **Service**.
4. In the **Setup and Maintenance** work area select the Service offering.

5. Click the **Opt In Features** button.

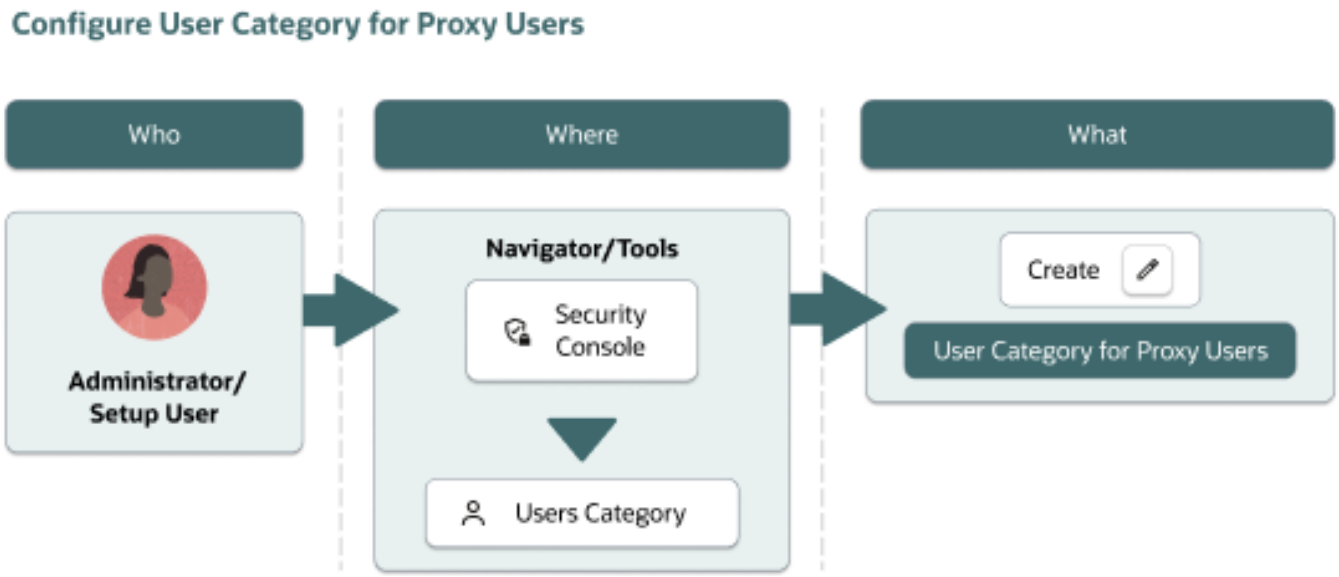
   The **Opt In: Service** page is displayed.
6. Find the **Digital Customer Service** row, then select the **Enable** check box.
7. Click **Done**.

Enabling Digital Customer Service will let you find these work areas grouped with the Service icon in Fusion Service:

- Self-Service Users
- Registration Requests

Configure a user category for proxy users

A proxy user account is used by multiple actual users to perform REST operations in Fusion Service.



You can configure the proxy users to have a different password expiry policy. You're strongly encouraged as it ensures that the regular password expiry rules don't apply to proxy users. You can create a different user category for the proxy users.

Follow these steps to create a category specific for proxy users that uses a password policy that doesn't expire passwords:

1. Sign in to Fusion Service.

   On the home page, select Tools, then **Security Console**.
2. In the **Security Console** work area, click the **User Categories** tab.
3. Click **Create**.

   The **User Category: Details** page appears.

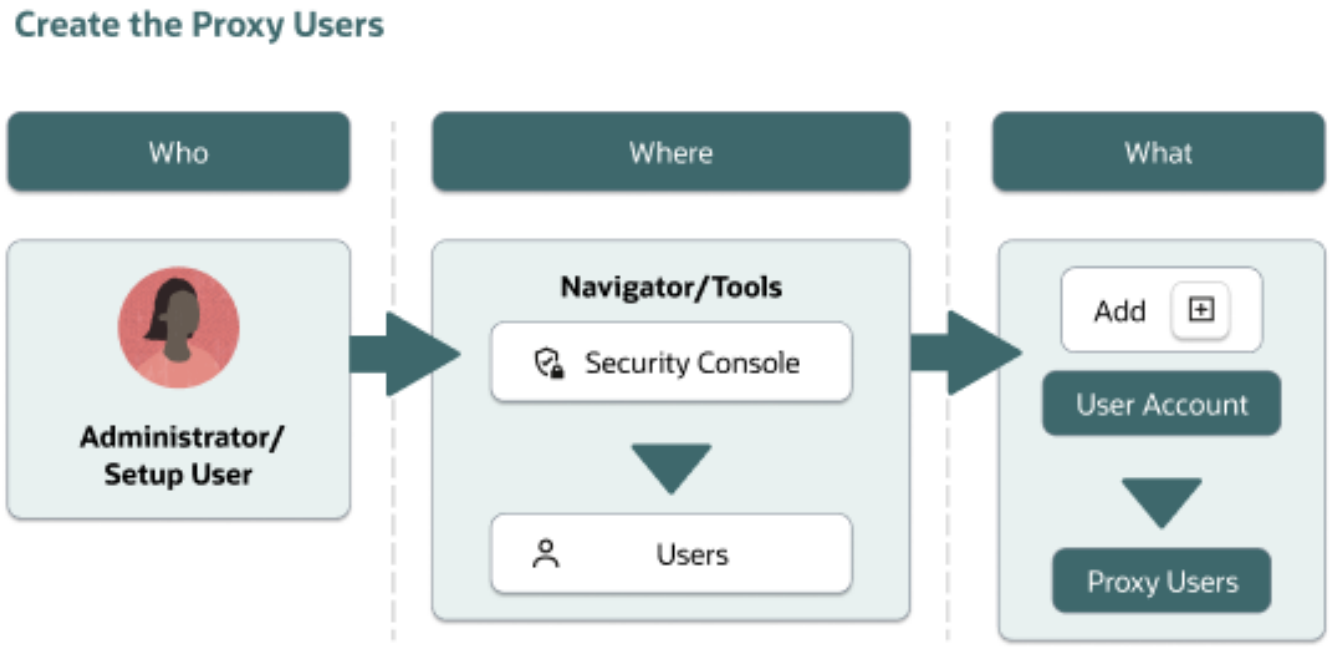
4. Click **Edit**.
5. Enter a name in the **User Category Name** field, such as **ProxyUsers**.

   > **Note:** Note this field can't contain spaces or special characters.

6. Click **Save and Close**.
7. Click **Password Policy**.
8. Click **Edit**.
9. Configure the fields for the proxy user password policy. In particular, you might want to set the **Days Before Password Expiration** to a relatively high number and give enough time for the **Days Before Password Expiry Warning**.
10. Click **Save and Close**.
11. Click **Done**.

Create the proxy users

A proxy user account is used by multiple actual users to perform REST operations in Fusion Service.



We recommend you use the syntax shown in the following table when you create your proxy user accounts.

1. Sign in to Fusion Service as an administrator or setup user.
2. Click Navigator, then from the Tools menu, select **Security Console**.
3. In the **Security Console** work area, click the **Users** tab.
4. Click **Add User Account**.
5. Now, one at time, add three separate users and grant them roles by doing the following:

a. From the **User Category** drop-down list, select **ProxyUsers**.
b. Fill in the **First Name**, **Last Name**, and **Email** fields.

> **Note:** The **User Name** field auto fills with the email address user name. Overwrite that in the following step.

c. In the **User Name** field, choose one of the entries:

User	Roles Granted	Role Code
PUDS_CSS_USER	Customer Self-Service Proxy User	ORA_SVC_CUSTOMER_SELF_SERVICE_PROXY_USER_ABSTRACT
PUDS_CSS_ADMIN	Customer Self-Service Proxy Account Administrator	ORA_SVC_CUSTOMER_SELF_SERVICE_PROXY_ACCOUNT_ADMINISTRATOR
PUDS_ANONYMOUS_USER	Customer Self-Service Proxy Anonymous User	ORA_SVC_CUSTOMER_SELF_SERVICE_PROXY_ANONYMOUS_USER

d. Enter a password, then confirm the password.

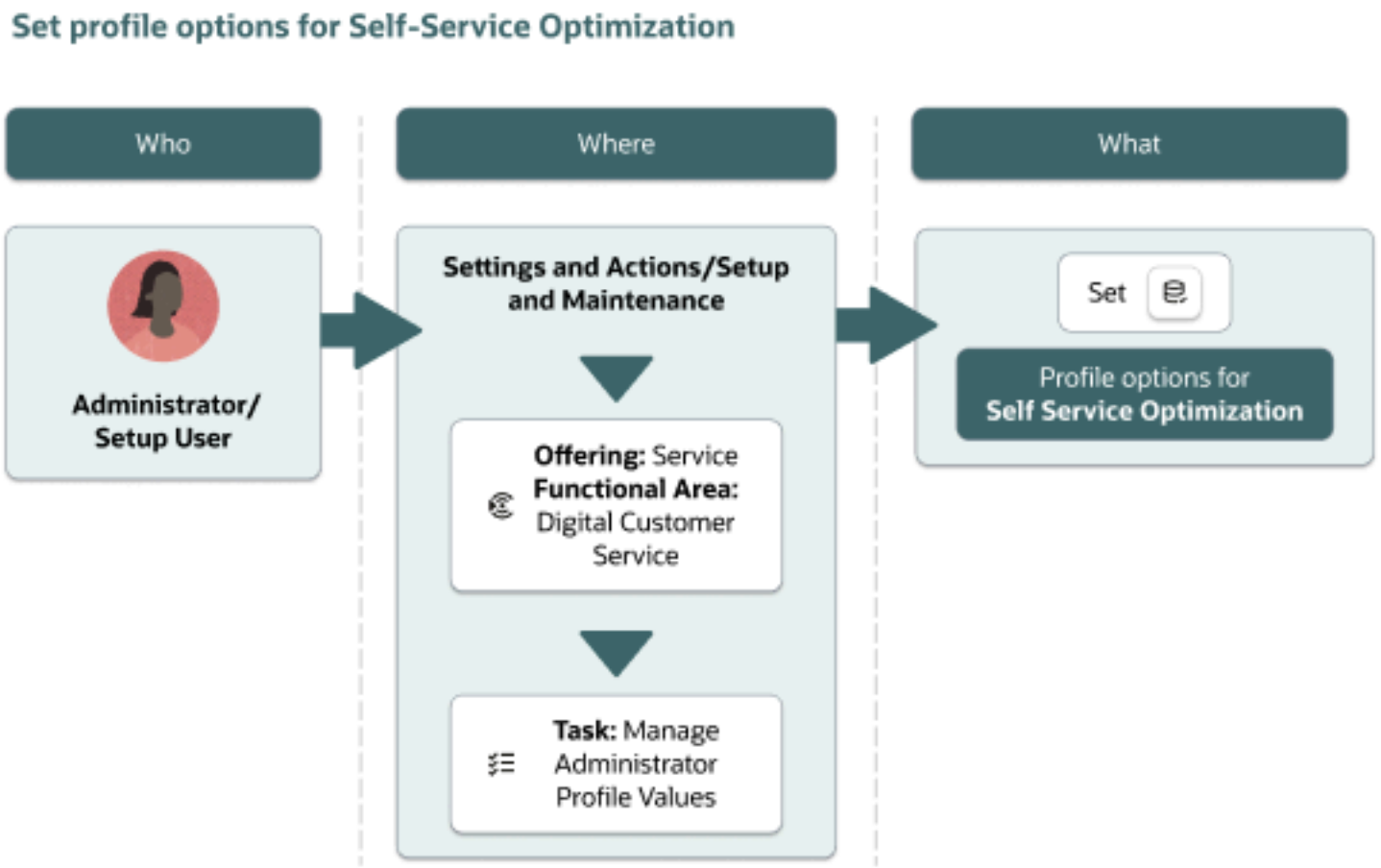
> **Note:** Passwords must be at least eight characters long.

e. Click **Add Role**.
f. In the Search field, enter the value listed in the **Roles Granted** column of the following table that corresponds with the user you've created.
g. When the role is found, click **Add Role Membership**, and then click **Done**.
h. Click **Save and Close** and repeat these steps for each of the three users you must create.

How do I set profile options for Self-Service Optimization?

Self-Service Optimization gives you greater flexibility by using proxy users in Fusion Service

You must set the required profile options for Self-Service Optimization. There are also optional profile options for other functionality.



Set Required Digital Customer Service Profile Options

The profile options specific to Digital Customer Service are found in two task areas: Manage Digital Customer Service Profile Options and Manage Digital Customer Service Account Setup Profile Options.

1. Sign in to Oracle Fusion Service as administrator or a setup user.
2. In the **Setup and Maintenance** work area, go to the following:
    - Offering: Service
    - Functional Area: Digital Customer Service
    - Task: Manage Digital Customer Service Profile Options

      or
    - Task: Manage Digital Customer Service Account Setup Profile Options
3. Click the name of the profile option that you want to change.
4. Set the profile option value as needed.
5. Click **Save and Close**.

Required Profile Options and Descriptions	Default Value	Comments
FND_IDP_PROXY_USER_WHITELIST Used to identify the list of allowed proxy users.	None	Enter a comma-separated list of proxy user names. **Note:** There must be no spaces between the commas and the names.
ORA_CORS_ORIGINS List of trusted domains that can make requests.	None	*, or specific comma-separated fully qualified domain names.
CORS_ACCESS_CONTROL_ALLOW_HEADERS Specifies comma-separated list of headers that are allowed as part of a CORS request.		Add these values, in the comma separated list, if they're not present: Puds-Access-Token, kmauthtoken, content-language, X-Oracle-ABCS-SessionId, X-Oracle-ABCS-UserId
SVC_CSS_PUDS_CACHE_DURATION Decides the amount of time, in minutes, that Proxy User Data Service objects are cached.	15 minutes	Any changes to this parameter will force a refresh of the proxy users configuration data cache.
SVC_CSS_USE_FA_AS_IDP Identifies if self-service users are created in Fusion Service or in IDCS.	False.	Make sure this value is set to False for self-service optimization mode.

Set Optional Profile Options

Profile options enable you to configure and control application data centrally. They store user preferences, installation information, configuration choices, and processing options. Administrators and setup users manage profile options in the Setup and Maintenance work area.

Registration Profile Options

Profile Option	Default Value	Possible Values	Effect
`SVC_CSS_SELF_REGISTRATION`	New Or Existing	None New Or Existing Existing Only	Specifies which contacts can self-register. If Existing is specified, only existing contacts can self-register.
`SVC_CSS_SELF_REG_AUTO_APPROVE`	False	True False	Enables automatic approval of self-service registration requests that are associated with an account. If SVC_ CSS_ SELF_ REG_AUTO_ APPROVE is set to False and SVC_ CSS_ ACCT_ADMIN_APPROVE is set to True, then the Digital Customer Service Account

Profile Option	Default Value	Possible Values	Effect
			Administrators can approve user registration requests in the Digital Customer Service Customer user interface. Also, Digital Customer Service Administrators can approve registration requests in the Digital Customer Service Administration user interface.
SVC_CSS_ACCT_ADMIN_APPROVE	True	True False	Enables the approval of self-service user registration requests by users with Digital Customer Service Account Administrator roles. If set to True, Digital Customer Service Account Administrators can approve user registration requests in the Digital Customer Service customer user interface and Digital Customer Service Administrators can approve registration requests in the Digital Customer Service Administration user interface. If set to False, only Digital Customer Service Administrators can approve registration requests in the Digital Customer Service Administration user interface. **Note:** This option applies only when the SVC_ CSS_ SELF_ REG_ AUTO_APPROVE option is set to False.
SVC_CSS_ACCT_KEY_FIELD	OrganizationName	Any field in the Account object	Specifies a valid field name in the Account object. The field name is case sensitive. **Note:** You must create an attribute in the account object to be the account key, because the default account key of account name isn't secure.
SVC_CSS_REG_CONT_MAP	An empty string	An empty string. Any defined value, with a colon separating fields, and commas separating the pairs. For example, reg_field1:contact_ field1, reg_field2:contact_field2	You specify a value for this profile option only if the name of the attribute in the Self Registration object is different from the name in the Contact object. Cases where they might happen are if you've created a custom attribute for an object. Custom attributes are

Profile Option	Default Value	Possible Values	Effect
			designated with an _c, such as PlaceOfBirth_c. For this use case, you ignore the _c when decide whether an attribute maps or not. So, let's take the custom attribute in the Self Registration object **`PlaceOfBirth_c`**. Because the Contact object has a ready to use attribute called **`PlaceOfBirth`**no mapping is required because the two values match. If, however, the name of the custom attribute was **`BirthPlace_c`** the value of this profile option would then be **`BirthPlace_c:PlaceOfBirth`**. Here's an extra example with multiple mappings: First, you specify case sensitive name and value pairs to map the fields of the Registration View object to the Contact View object in the following way: **`reg_field1:contact_field1,reg_field2:contact_field2`**. The **`reg_field1`** is the PlaceOfBirth_c in the Registration View Object which is a custom object created in Application Composer. The **`contact_field`** is the **`PlaceOfBirth`** field in the Contact View object. This attribute is already present in the Contact object. So the mapping would be: **`reg_field1:contact_field1 LIKE BirthPlace_c:PlaceOfBirth`**
`SVC_CSS_SIGN_IN_ATTR_NAME`	EmailAddress	The value of the assigned attribute must be unique. Possible values include: EmailAddress	Specifies the sign-in attribute that users must specify in the Sign in ID field in the Self-Service Registration object. This field is used to decide whether the user exists in the Lightweight Directory Access Protocol server.
`SVC_CSS_REG_FLD_CONTACT`	EmailAddress	Any field on the Self-Service Registration object.	Specifies the field to use during the user registration process to decide if the registering user is an existing contact. The field names are case sensitive.

Profile Option	Default Value	Possible Values	Effect
			The SVC_CSS_REG_CONT_MAP profile option is used to find the name of the attribute on the Contact.
SVC_CSS_SEND_WELCOME_EMAIL	True	True False	Enables sending a welcome email when a new user account is created.
SVC_CSS_USER_ROLE_COMMON_NAME	ORA_SVC_CUSTOMER_SELF_SERVICE_USER_ABSTRACT	A string representing the name of the role that's set up for Customer Self-Service users. Typically, this is a copy of a Customer Self-Service User with extra privileges added.	Specifies the common name of the role granted to previously created Customer Self-Service Users.
SVC_CSS_ACCT_ADMIN_ROLE_COMMON_NAME	ORA_SVC_CUSTOMER_SELF_SERVICE_ACCOUNT_ADMINISTRATOR_ABSTRACT	A string representing the name of the role that's set up for Customer Self-Service Account Administrator. Typically, this is a copy of a Customer Self-Service Account Administrator with extra privileges added.	Specifies the common name of the role granted to the previously created Customer Self-Service Account Administrators.
SVC_CSS_USER_CATEGORY	An empty string	A string	Specifies the user category that defines the URL to which the self-service user is redirected after a password reset. The user category is defined in the Security Console.
SVC_CSS_IMP_SIGN_IN_ATTR_NAME	PrimaryEmailAddress	Any field on the Contact object.	Specifies a field in the Contact object to be used as the sign-in attribute when importing data into the Self-Service Roles object. The field name is case sensitive.
SVC_CSS_USE_FA_AS_IDP	False	True False	Specifies whether the identity provider is Oracle Fusion Applications or Oracle Identity Cloud Service. When set to True, Oracle Fusion Applications is used.
ORA_SVC_CSS_SELF_REG_B2C_AUTO_APPROVE	True	True False	Enables automatic approval of self-service registration requests that aren't associated with an account. If set to True, users who register without an account will be auto approved to become self-service users. If set to False, users who register without an account

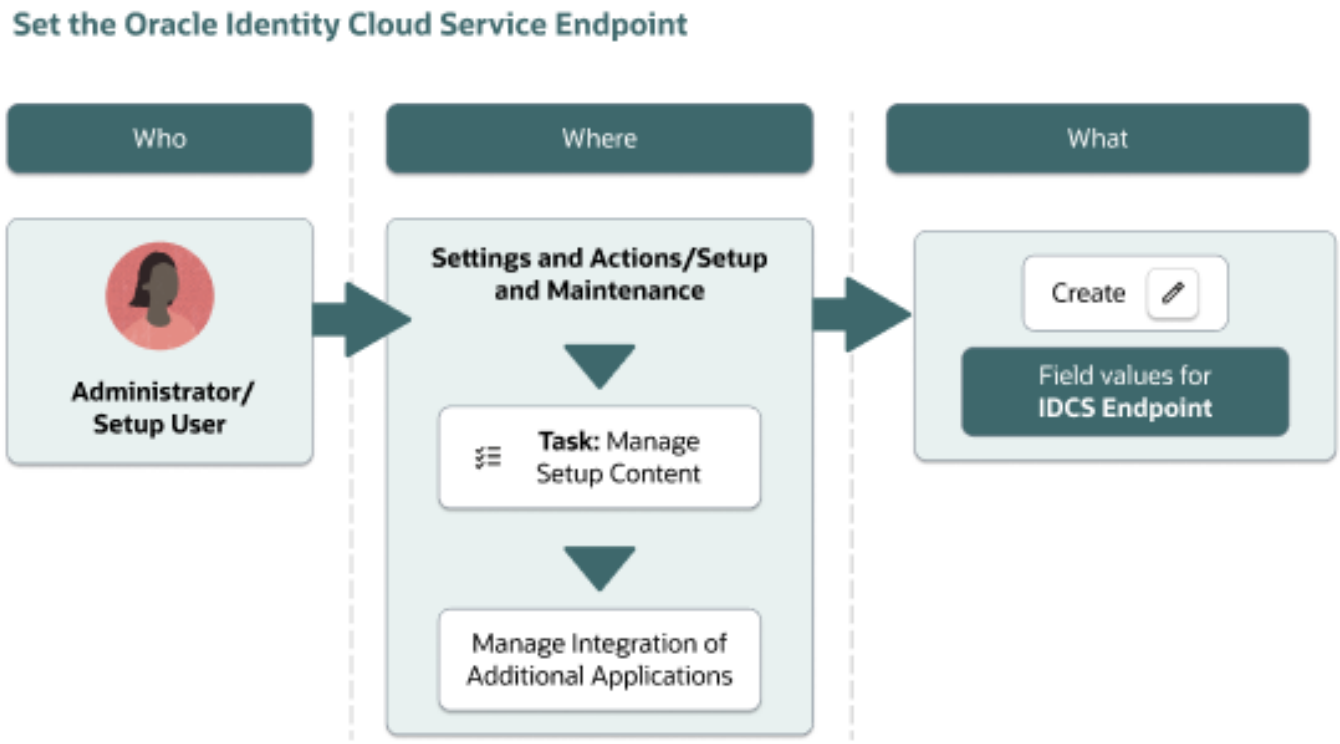
Profile Option	Default Value	Possible Values	Effect
			will need to be approved by an administrator before they can become self-service users.
SVC_CSS_PUDS_CACHE_ DURATION	15	Integer in minutes	Decide the amount of time, in minutes, that Self-Service Optimization objects are cached.
SVC_CSS_ALLOW_CONTACT	True	True False	Enables the self-service registration of B2C Service contacts.
SVC_CSS_ALLOW_CONSUMER	True	True False	Enables the self-service registration of consumers.
SVC_CSS_CONSUMER_USER_ CATEGORY	An empty string	A string	Specify the user category for consumers defining the redirect URL for self-service users after a password reset.

Knowledge Setup Profile Options

Profile Option	Default Value	Possible Values	Effect
CSO_CONTENT_RATING_TYPE	None	True and False	Enables content rating for Knowledge.

Set the Oracle Identity Cloud Service endpoint

You can configure Fusion Service to enable back end communication with IDCS.



Here are the steps:

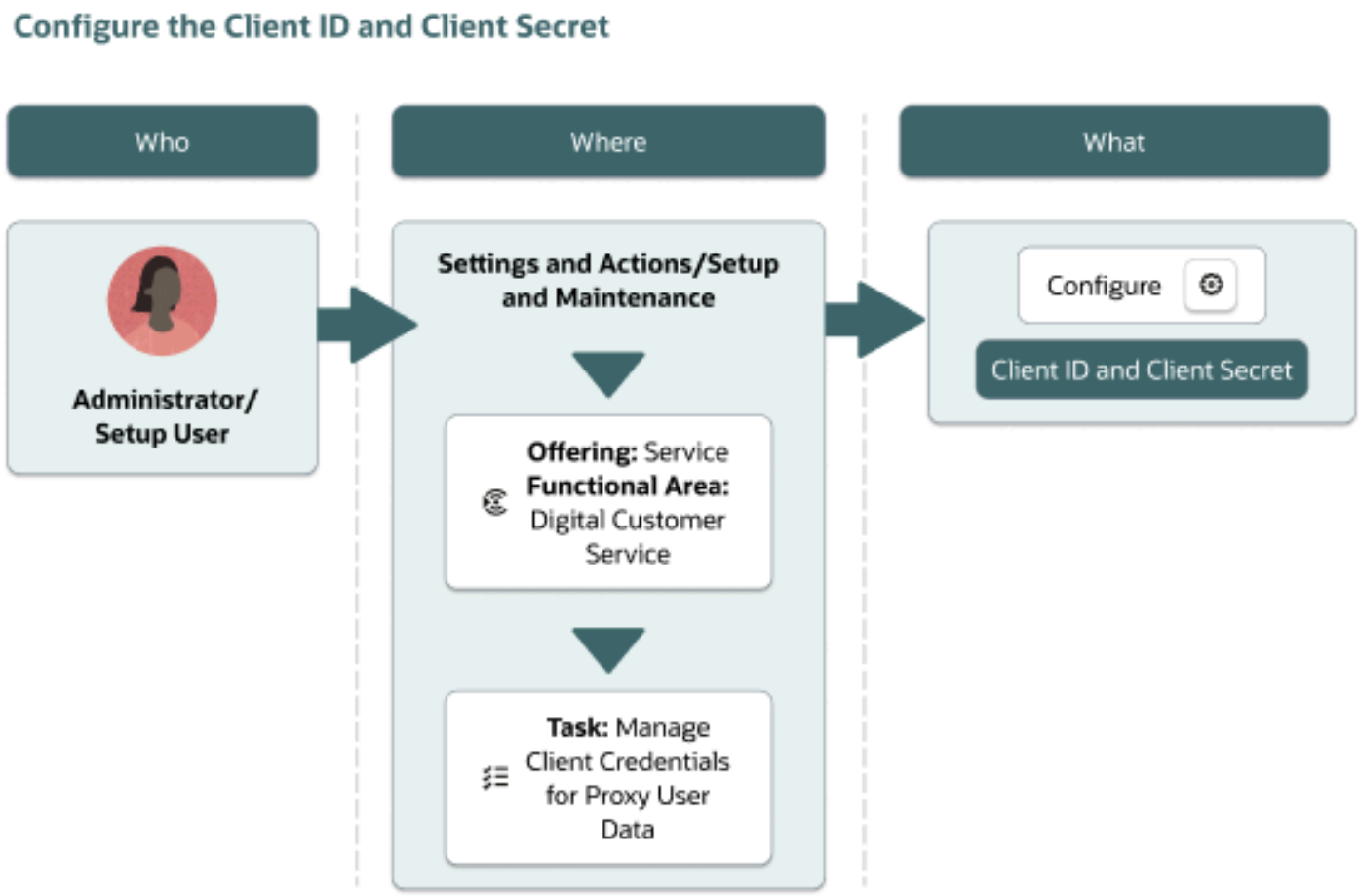
1. Sign in to Fusion Service as administrator or setup user.
2. Click the Settings and Actions menu, and select Setup and Maintenance.
3. Click the **Tasks** icon, then click **Manage Setup Content**.
4. Click **Manage Integration of Additional Applications**.
5. In the Search Results area, click the **Actions** menu, and select **Create**.
6. In the Create Application Integration form, enter the following:

Field	Enter this value
Application Name	IDCS_REST_ENDPOINTAPP
Full URL	For example: `https://<IDCS HOST>/admin/v1` **Note:** This is the IDCS host that authenticates the Visual Builder instance.
Partner Name	IDCS

7. Click the **Apply** button and then verify the Protocol, Server Host and Context Root details are populated in the **Server Details** area.
8. Click **Save and Close**.

Configure the client ID and client secret

Add the client ID and client secret that you created using the steps from *Create a Client application* to your Fusion Service configuration.



Follow these steps to configure the client ID and client secret.

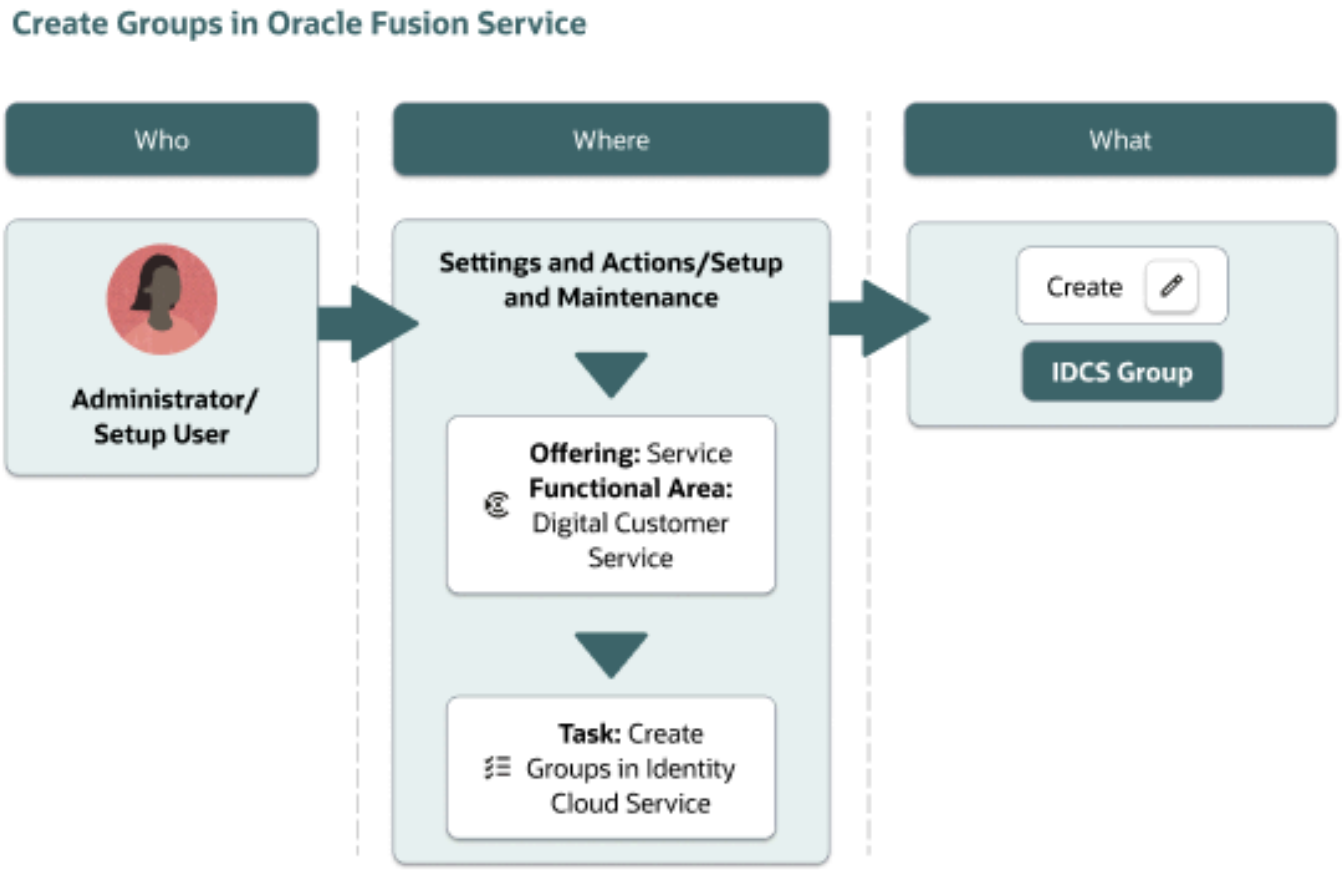
1. Sign in to Fusion Service as an administrator or a setup user.
2. In the Setup and Maintenance work area, go to the following:
    - Offering: Service.
    - Functional Area: Digital Customer Service.
    - Task: Manage Client Credentials for Proxy User Data
3. Enter the Client ID and Client Secret.

4. Click **Test** to make sure the values you entered are valid, and then click **OK** if so.
5. Click **Save and Close**.

Note: The credentials are cached by the Proxy User Data Service. These credentials can be cleared or refreshed by changing the value of SVC_CSS_PUDS_CACHE_DURATION profile option.

Create groups in Fusion Service

The groups should already be in place in the Identity Cloud Service from the synchronization of roles from Fusion. If they're not, you can create them manually by working through the following steps.



Note: If you delete a group that's in use and then at another time recreate it we recommend that you clear the cache by adjusting the value of SVC_CSS_PUDS_CACHE_DURATION profile option.

Role Name	External ID	Related Profile Option
Customer Self-Service User	ORA_SVC_CUSTOMER_SELF_SERVICE_USER_ABSTRACT	SVC_CSS_USER_ROLE_COMMON_NAME
Customer Self-Service Account Administrator	ORA_SVC_CUSTOMER_SELF_SERVICE_ACCOUNT_ADMINISTRATOR_ABSTRACT	SVC_CSS_ACCT_ADMIN_ROLE_COMMON_NAME

Note: If you've changed the value of the profile options listed in the last column of the following table, then don't use the external ID and role name shown in the table, instead the external ID and role name will depend on the value of your change profile option.

Note: By default, the related profile options reflect the preceding table. If you use custom roles, the Role Name and the Display Name values must match the value of the relevant profile option. If you need to create a new group, you do it using the **Create Groups in Identity Cloud Service** task in Functional Setup Manager. Here are the steps for you to create groups:

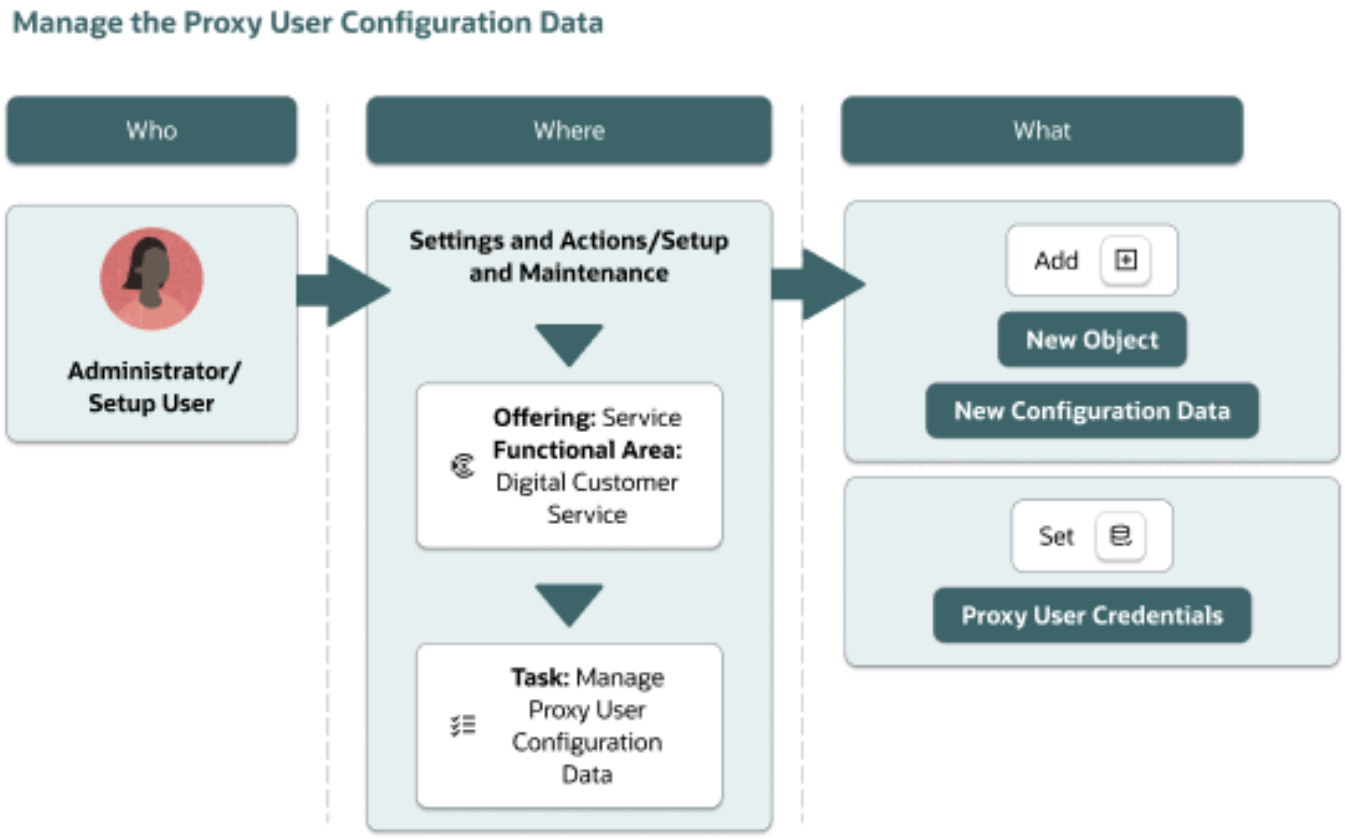
1. Sign in to Fusion Service as an administrator or setup user.
2. In the Setup and Maintenance area, go to the following:
   - Offering: Service.
   - Functional Area: Digital Customer Service.
   - Task: Create Groups in Identity Cloud Service.

     Note: Select **All Tasks** from the **Show** drop-down list to display the task.
3. Click the link for the task.
4. Select the appropriate Fusion Service role from the table.
5. If the corresponding Identity Cloud Service role doesn't exist, click the **Create** button to create the IDCS group.

   Note: If the corresponding IDCS group already exists, you'll see information about the existing role instead of the Create button.

Manage the proxy user configuration data

Perform these steps to manage all the URL patterns that you need to reach in Fusion Service through the proxy user data service.



The fields are pre populated in Fusion Service with all the endpoints that are used in the Digital Customer Service reference implementation. The standard use case it to set up the proxy user credentials.

If you need to add more objects, here are the steps:

1. Sign in to Fusion Service as administrator or setup user.
2. In the Setup and Maintenance work area, go to the following:
    - Offering: Service.
    - Functional Area: Digital Customer Service.
    - Task: Manage Proxy User Configuration Data.
3. For predefined configuration data note the following:
    - You can add new configuration data.
    - Only allowlisted resources can be exposed to anonymous users. The following resources are on the allowlist:
        - categories
        - catalogProductItems
        - catalogProductGroups

        - productGroupHierarchies
        - fndStaticLookups
        - selfRegistrations
        - chatAuthenticate
        - dynamicLinkPatterns
    - Custom objects that you create with Application Composer can be exposed to Anonymous users even though they're not on the allowlist.
4. To add new objects, do the following:
    a. In the Search Results area, click the **New (+)** icon.
    b. Add the URL pattern.
    c. Click the drop-down list and add the HTTP method.
    d. Click the drop-down list and choose the appropriate user role.

    > **Note:** This is the same user role from Identity Cloud Service.

    e. Add the appropriate proxy user key.
    f. Click **Active**.
5. Click **Save and Close**.

> **Note:** Proxy user configuration data is cached for a duration specified by the SVC_CSS_PUDS_CACHE_DURATION profile option.

If this value is changed then the cache is cleared. For changes made using this UI to be read immediately, the value of SVC_CSS_PUDS_CACHE_DURATION profile must be altered. The recommended action is to add or subtract 1 minute from the existing value.

How do I set proxy user credentials in DCS?

Complete the configuration of the proxy user configuration data for the Self-Service Optimization feature, you need to set the user credentials for the proxy users.

1. Sign in to Fusion Service as an administrator or setup user.
2. In the Setup and Maintenance work area, go to the following:
    - Offering: Service.
    - Functional Area: Digital Customer Service.
    - Task: Manage Proxy User Configuration Data.
3. Find any line referencing the following proxy user keys:
    - PUK#_ANONYMOUS_USER
    - PUK#_SELF_SERVICE_USER
    - PUK#_SELF_SERVICE_ACCOUNT_ADMIN

4. Do the following with each of the three user keys:
    a. Select the entry, then from the **Actions** menu, click **Edit**.
    b. In the Edit Proxy User Configuration Data workspace, enter the Proxy User Name if it's different from the default name.

        > **Note:** It's recommended that you retain the proxy user names.

    c. Enter the password for the proxy user.
    d. Click **Save and Close**.

Because the same proxy user key is used by multiple URLs, you can edit any one URL that uses this proxy user key to set the proxy user credentials associated with that key.

> **Note:** Proxy user configuration data is cached for a duration specified by the SVC_CSS_PUDS_CACHE_DURATION profile option.

Validate Self-Service Optimization setup

Now that you've enabled Fusion Service, you can verify the setup.

You can use a set of validation checks Functional Setup Manager by providing certain basic configuration parameters. The checks are grouped by task and have a set of tests associated with them. Some tests requires you to enter parameters. If so, the Supply Parameters tab is enabled. The Test Results tab is enabled once all required parameters have been populated.

Here's an overview of all the tests you can run:

Validate Self-Service Optimization Tests

Setup Task	Tests	Description
Set Profile Options for Self-Service Optimization	Verify profile option: SVC_CSS_USE_FA_AS_IDP	Verifies the value of the SVC_CSS_USE_FA_AS_IDP profile option.
	Verify profile option: FND_IDP_PROXY_USER_WHITELIST	Verifies the value of the FND_IDP_PROXY_USER_WHITELIST profile option.
	Verify profile option: ORA_CORS_ORIGINS.	Verifies the value of the ORA_CORS_ORIGINS profile option.
	Verify profile option: CORS_ACCESS_CONTROL_ALLOW_HEADERS	Verifies the value of the CORS_ACCESS_CONTROL_ALLOW_HEADERS profile option.
Configure the Client ID and Client Secret	Get IDCS client access token	Retrieves the client access token for use in other tests.
Create Groups in Oracle Identity Cloud Service	Verify customer self-service user group in IDCS	Verifies that the Customer Self-Service User group is available in Identity Cloud Service.

Setup Task	Tests	Description
	Verify customer self-service account administrator group in IDCS	Verifies that the Customer Self-Service User group is available in Identity Cloud Service.
Set Proxy User Credentials	Get IDCS user access token	Retrieves the user access token for use in other tests.
	Verify IDCS and Fusion Service integration	Verifies the Identity Cloud Service and Fusion Service Integration.
	Verify anonymous proxy user setup	Verifies the Anonymous Proxy User setup.
	Verify customer self-service proxy user setup	Verifies the Customer Self-Service Proxy User setup.
	Verify customer self-service proxy account administrator setup	Verifies the Customer self-service proxy account administrator setup
	Get categories as anonymous user	Verifies the categories are available to an anonymous user

Run pre validation setup in IDCS

1. Sign in to IDCS as the Administrator user.
2. Click the Navigation drawer and then expand the **Applications** list.
3. Search for **Proxy User Data Service App** and select it.
4. Select the **Configure** tab, and expand the **Client Configuration** menu item.
5. In the Authorization area, select the **Resource Owner** along with the already selected **Client Credentials** option for Allowed Grant Types.
6. Click the **Test Results** step.

Run validations tests in Fusion Service

1. Sign in to Fusion Service as an administrator or setup user.
2. In the Setup and Maintenance work area, go to the following:
   - Offering: Service.
   - Functional Area: Digital Customer Service.
   - Task: Validate Self-Service Optimization Setup for Digital Customer Service.

   > **Note:** To show this task, you might need to select **All Tasks** from the **Show** drop-down list.
3. On the **Select Test** page, select the tests that you'd like to run.

   If a given test request requires parameters, the **Supply Parameters** step will be enabled.
4. If required, enter parameters in the **Value** field on the **Supply Parameters** page.

   > **Note:** The **Supply Parameters** step is only enabled if parameters are required for the chosen test.
5. Click the **Test Results** step.

The tests are automatically run and the results displayed. You can return to the **Select Test** page as often as necessary until you've fully performed your validations.

Run post validation tests in IDCS

1. Sign in to IDCS as the Administrator user.
2. Click the Navigation drawer and then expand the **Applications** list.
3. Search for **Proxy User Data Service App** and select it.
4. Select the **Configure** tab, and expand the **Client Configuration** menu item.
5. In the Authorization area, deselect the **Resource Owner** option for Allowed Grant Types.

Set Up Administrators and Developers

What are the Digital Customer Service Developer roles?

To work with business objects relevant to Digital Customer Service, the developer must be a Digital Customer Service user and must be granted the appropriate roles. Here are thee tasks you'll need to perform:

1. Create an Internal Customer Account (This task only needs to be done once). *How do I create an Internal Customer account?*
2. Register the Developer as a Self-Service User. *How do I create an Internal Customer account?*
3. Add User Roles. *How do I add Visual Builder roles?*

How do I create an Internal Customer account?

Before creating the Digital Customer Service developer or administrator users, an internal customer account must be created for these users.

The internal customer account lets you associate your staff to a specific account. An account key is required when creating a new Digital Customer Service user.

> **Note:** You only need to create the internal customer account once. This internal customer account can be used by all Developers and Administrators.

To create an internal customer account:

1. Sign in to Oracle Fusion Service.
2. Navigate to the **Service** work area and click **Accounts**.
3. Click **Create Account**.
4. Enter the **Name**.

> **Note:** The name that you enter in this field represents the **AccountKey** that you need to use in the Create Digital Customer Service Developers topic. This is true only if the value of the profile option SVC_CSS_ACCT_KEY_FIELD hasn't been changed.

5. Select **Customer** from the **Type** menu.
6. Click **Save and Close**.

Create Digital Customer Service Developers

Here's how you create Digital Customer Service Developer users.

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Click the Settings and Actions menu, and select **Setup and Maintenance**.
3. In the **Setup and Maintenance** work area, go to the following:
    - Offering: Service
    - Functional Area: Digital Customer Service
    - Task: Create Digital Customer Service Developers

    > **Note:** To access this task, click the **Show** drop-down list, and select **All Tasks**.
4. In the Create Developers for Digital Customer Service page, fill in each field.

    The default value of the account key is the account name you created in *How do I create an Internal Customer account?* . You're encouraged to make the account key more secure by creating a custom account key here: *Create a custom Account Key field*.

    > **Note:** The sign in ID is optional. To use your email address as the sign in ID, just click the check box.
5. Click **Save** when you're finished.

    The user account is created in both Fusion Service and IDCS. You'll then receive an email from IDCS prompting you to reset your password.

Set Up Oracle Visual Builder

Retrieve the Oracle Visual Builder URLs

You must have access to Oracle Visual Builder to perform setup tasks for Visual Builder.

Use these steps to find URLs for:

- Oracle Visual Builder. Use for your application development and its lifecycle. VBCS has a 1:1 mapping with the Fusion Service Production, Test and Development environments.
- Oracle Visual Builder Studio (optional). You can use Visual Builder Studio for source control and to build pipelines. You can integrate VB Studio with Visual Builder.

> **Note:** Only 1 instance of VB Studio is provisioned for each Cloud Account and is associated with the **Test Fusion Service** instance.

1. Sign in to Fusion Service as an administrator or setup user.

2. In the Setup and Maintenance work area, go to the following:

- Offering: Service.
- Functional Area: Digital Customer Service.
- Task: View Digital Customer Service URLs.

Note: To view the task, select **All Tasks** from the **Show** drop-down list.

1. In the View Digital Customer Service URLs page, copy the URLs by finding the entry or either:
   a. Oracle Visual Builder
   b. Oracle Visual Builder Studio
2. When you've found the entry you want, click the **Copy** button.

Specify the Fusion Service Details in Oracle Visual Builder

To specify Fusion Service details in Oracle Visual Builder:

1. Sign into Oracle Visual Builder as an administrator.
2. Click the **Menu** icon, and select **Settings** to open the Tenant Settings page.
3. Click the Services tab, then click the **Back ends** icon (**+**)
4. In the Back end Service Type window, select **Oracle Cloud Application Instance**.
5. In the Instance URL field of the Create Oracle Cloud Application Instance window, enter the instance URL of your Fusion Service back end service.
6. From the Authentication drop-down list, select **Oracle Cloud Account**.
7. Click **Create**.

Note: You must set the Instance URL field with the fully qualified domain name of your Oracle Applications Cloud Fusion Service instance.

Verify Your Oracle Visual Builder Settings

Here's how to verify your Oracle Visual Builder settings:

1. Sign in to Oracle Visual Builder as a Service Administrator.
2. Click the **Home** menu.
3. Click **Settings**.

   The **Tenant Settings** screen appears.
4. On the **General** tab, find the **Allow only secure applications to be created** option, and ensure that it's turned off.

4 Create a Basic Digital Customer Service Application

Create a New Digital Customer Service application

The steps that follow assume that you've selected the current Reference Implementation template.

1. Sign in to the Oracle Visual Builder editor as a user with the Developer role.
2. Click **New**.

   > **Note:** If no applications have been created yet, the button will read **New Application**.

   The **Create Application** dialog appears.
3. In the **Application template** section, click **Change template**.
4. Click the newest reference implementation tile which will match the Fusion Service version you're using.

   This template creates an application with basic service functionality, including the ability to create and update SRs, search knowledge, and chat with an agent.
5. Click **Select**.
6. Specify the **Application Name**.
7. Specify the **Application ID**.

   The **Application ID** is automatically derived from the specified Application Name, but you can change it, if desired. The Application ID can't be changed once the application is created: it appears in the application URL.
8. Click **Finish**.

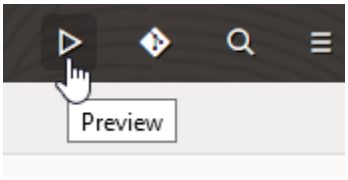
   Once the Digital Customer Service application has been created, you're now ready to configure the application to meet your business needs.

> **Note:** When Oracle Fusion Service is deployed with multiple business units, more configuration is required.

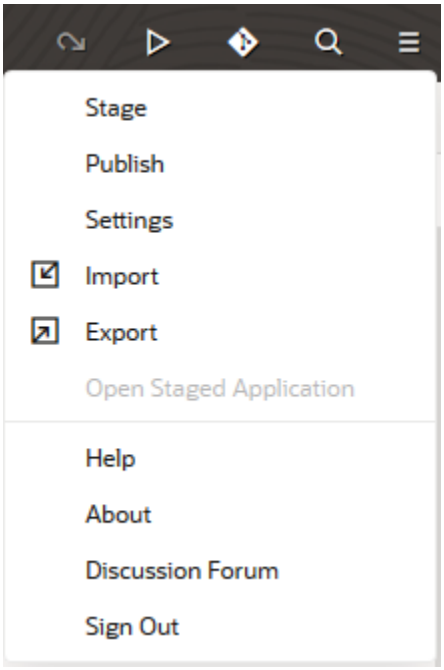
> **CAUTION:** For your Digital Customer Service Reference Implementation template to work properly, access to Knowledge and Chat must be allowed anonymously. Your Oracle Visual Builder settings must to be configured to allow publicly accessible applications to be created. To verify the settings, do the following:
>
> 1. Open the application artifact and select the Settings editor in the Designer.
> 2. Open the Security tab and then, in the Access pane, select **Allow anonymous access**.

Stage and Publish the Digital Customer Service Application

You can preview your changes as you develop them by clicking the **Preview** button.



Test that your changes perform as expected by staging the application. Access the **Stage** option by clicking the **Menu** button from the Visual Builder home page as shown in the following example:



Staging your application enables you to test each update you make to confirm that it acts as you expect and that no problems have been introduced, for example, when you add new features or change your data model.

When you publish a version of your application, it becomes read-only and can no longer be changed. To make changes or updates to your application, you need to create a new version. When you publish a staged version of your application, it becomes the active version.

> **Note:** After staging and publishing the application you must use the Self-Service user. The Developer user is only applicable when the application is in Preview mode.

If you're updating an earlier version of your application, the previous active version is archived and locked. Only one version of an application can be active at a time, but multiple versions of an application can be staged simultaneously.

5 Develop Your Digital Customer Service Application

Overview of Developing Your Digital Customer Service Application

The Oracle Digital Customer Service reference implementation is an application built with the Oracle Visual Builder Visual Applications platform.

The reference application come configured with pages, styling, and functionality supporting a typical self-service experience. The Visual designer enables developers to make changes and implement the look-and-feel of a brand.

As with any development framework, there can be restrictions when using ready-to-use components and capabilities. While Oracle Visual Builder is flexible and code can be written to accommodate many use cases, Oracle recommends using the packaged components as a first fulfillment of your requirements. If those components don't meet your requirements, then you may extend the capabilities.

> **Note:** If you have modified your Digital Customer Service deployment, you must devise a product life cycle strategy to manage your own code migration and merges as well as uptake any environment changes.

Your Digital Customer Service application has rules for business objects to execute business logic that deals with the data. Using object and field validators you can ensure data at the field or record level is correct. For more information, see the Related Topics section.

How do I configure DCS profile options?

This topic describes Digital Customer Service profile options, including configuration instructions.

The following two task lists apply to profile options for Digital Customer Service:

- Manage Digital Customer Service Profile Options
- Manage Digital Customer Service Account Setup Profile Options

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, including the following:

- User preferences
- Installation information

- Configuration choices
- Processing options

Here we focus on Oracle Fusion Service profile options specific to Digital Customer Service.

Registration Profile Options

The following table lists the profile options for Digital Customer Service Registration Profile Options.

Profile Option	Default Value	Possible Values	Effect
SVC_CSS_SELF_REGISTRATION	New Or Existing	None New Or Existing Existing Only	Specifies which contacts can self-register. If Existing is specified, only existing contacts can self-register.
SVC_CSS_SELF_REG_AUTO_APPROVE	False	True False	Enables automatic approval of self-service registration requests that are associated with an account. If SVC_ CSS_ SELF_ REG_AUTO_ APPROVE is set to False and SVC_ CSS_ ACCT_ADMIN_APPROVE is set to True, then the Digital Customer Service Account Administrators can approve user registration requests in the Digital Customer Service Customer user interface. Also, Digital Customer Service Administrators can approve registration requests in the Digital Customer Service Administration user interface.
SVC_CSS_ACCT_ADMIN_APPROVE	True	True False	Enables the approval of self-service user registration requests by users with Digital Customer Service Account Administrator roles. If set to True, Digital Customer Service Account Administrators can approve user registration requests in the Digital Customer Service customer user interface and Digital Customer Service Administrators can approve registration requests in the Digital Customer Service Administration user interface. If set to False, only Digital Customer Service Administrators can approve registration requests in the Digital Customer Service Administration user interface.

Profile Option	Default Value	Possible Values	Effect
			Note: This option applies only when the SVC_ CSS_ SELF_ REG_ AUTO_APPROVE option is set to False.
SVC_CSS_ACCT_KEY_FIELD	OrganizationName	Any field in the Account object	Specifies a valid field name in the Account object. The field name is case sensitive. **Note:** You must create an attribute in the account object to be the account key, because the default account key of account name isn't secure.
SVC_CSS_REG_CONT_MAP	An empty string	An empty string. Any defined value, with a colon separating fields, and commas separating the pairs. For example, reg_field1:contact_field1, reg_field2:contact_field2	You specify a value for this profile option only if the name of the attribute in the Self Registration object is different from the name in the Contact object. Cases where they may happen are if you have created a custom attribute for an object. Custom attributes are designated with an _c, such as PlaceOfBirth_c. For this use case, you ignore the _c when determining whether an attribute maps or not. So, let's take the custom attribute in the Self Registration object **PlaceOfBirth_c**. Since the Contact object has a out of the box attribute called **PlaceOfBirth**no mapping is required since the two values match. If, however, the name of the custom attribute was **BirthPlace_c** the value of this profile option would then be **BirthPlace_c:PlaceOfBirth**. Here's a additional example with multiple mappings: First, you specify case sensitive name and value pairs to map the fields of the Registration View object to the Contact View object in the following way: **reg_field1:contact_field1,reg_field2:contact_field2**. The **reg_field1** is the PlaceOfBirth_c in the Registration View Object which is a custom

Profile Option	Default Value	Possible Values	Effect
			object created in Application Composer. The **contact_field** is the **PlaceOfBirth** field in the Contact View object. This attribute is already present in the Contact object. So the mapping would be: **reg_field1:contact_field1 LIKE BirthPlace_c:PlaceOfBirth**
SVC_CSS_SIGN_IN_ATTR_NAME	EmailAddress	The value of the assigned attribute must be unique. Possible values include: EmailAddress	Specifies the sign-in attribute that users must specify in the Login ID field in the Self-Service Registration object. This field is used to determine whether the user exists in the Lightweight Directory Access Protocol server.
SVC_CSS_REG_FLD_CONTACT	EmailAddress	Any field on the Self-Service Registration object.	Specifies the field to use during the user registration process to determine if the registering user is an existing contact. The field names are case sensitive. The SVC_CSS_REG_CONT_MAP profile option is used to locate the name of the attribute on the Contact.
SVC_CSS_SEND_WELCOME_EMAIL	True	True False	Enables sending a welcome email when a new user account is created.
SVC_CSS_USER_ROLE_COMMON_NAME	**ORA_SVC_CUSTOMER_SELF_SERVICE_USER_ABSTRACT**	A string representing the name of the role that's set up for Customer Self-Service users. Typically, this is a copy of a Customer Self-Service User with additional privileges added.	Specifies the common name of the role granted to previously created Customer Self-Service Users.
SVC_CSS_ACCT_ADMIN_ROLE_COMMON_NAME	**ORA_SVC_CUSTOMER_SELF_SERVICE_ACCOUNT_ADMINISTRATOR_ABSTRACT**	A string representing the name of the role that's set up for Customer Self-Service Account Administrator. Typically, this is a copy of a Customer Self-Service Account Administrator with additional privileges added.	Specifies the common name of the role granted to the previously created Customer Self-Service Account Administrators.
SVC_CSS_USER_CATEGORY	An empty string	A string	Specifies the user category that defines the URL to which the self-

Profile Option	Default Value	Possible Values	Effect
			service user is redirected after a password reset. The user category is defined in the Security Console.
SVC_CSS_IMP_SIGN_IN_ATTR_NAME	**PrimaryEmailAddress**	Any field on the Contact object.	Specifies a field in the Contact object to be used as the sign-in attribute when importing data into the Self-Service Roles object. The field name is case sensitive.
SVC_CSS_USE_FA_AS_IDP	False	True False	Specifies whether the identity provider is Oracle Fusion Applications or Oracle Identity Cloud Service. When set to True, Oracle Fusion Applications is used.
ORA_SVC_CSS_SELF_REG_B2C_AUTO_APPROVE	True	True False	Enables automatic approval of self-service registration requests that aren't associated with an account. If set to True, users who register without an account will be auto approved to become self-service users. If set to False, users who register without an account will need to be approved by an administrator before they can become self-service users.
SVC_CSS_PUDS_CACHE_DURATION	15	Integer in minutes	Determine the amount of time, in minutes, that Self-Service Optimization objects are cached.
SVC_CSS_ALLOW_CONTACT	True	True False	Enables the self-service registration of B2C Service contacts.
SVC_CSS_ALLOW_CONSUMER	True	True False	Enables the self-service registration of consumers.
SVC_CSS_CONSUMER_USER_CATEGORY	An empty string	A string	Specify the user category for consumers defining the redirect URL for self-service users after a password reset.

Account Setup Profile Options

The following table lists the profile options for Digital Customer Service Account Setup Profile Options.

Profile Option	Default Value	Possible Values	Effect
CSO_CONTENT_RATING_TYPE	None	True and False	Enables content rating for Knowledge.

Set Digital Customer Service Profile Options

This topic describes how to set profile options for Digital Customer Service. The profile options specific to Digital Customer Service are found in two task areas: Manage Digital Customer Service Profile Options and Manage Digital Customer Service Account Setup Profile Options.

To find and set the Digital Customer Service profile options:

1. Sign in to Oracle Fusion Service as administrator or a setup user.
2. In the **Setup and Maintenance** work area, go to the following:
    - Offering: Service
    - Functional Area: Digital Customer Service
    - Task: Manage Digital Customer Service Profile Options

      or
    - Task: Manage Digital Customer Service Account Setup Profile Options
3. Click the name of the profile option that you want to modify.
4. Set the profile option value as needed.
5. Click **Save and Close**.

Configure Source Control for Applications

Configuring Git for source control of your application is optional.

1. In Visual Builder Studio, set up a Git repository.

    > **Note:** For groups that will be collaborating, when you're creating your project, select the **Shared** option when defining **Security**.

    > **Note:** It's recommended that you create a branch for each developer to enable your developers to work independently and then merge to the main branch when they want their changes to be shared.
2. In Oracle Visual Builder, integrate your application with a Git repository.

Secure Pages

Page security is controlled at the flow level. You must view the flow Metadata "{}" to view and update the security setting for the flow.

For more information see the Related Topics section for a link to the Secure the Application section of the Developing Applications with Oracle Visual Builder in Oracle Integration guide.

Here's an example of how it works. In the reference implementation app, the contact-us flow is available anonymously, and has the following security metadata:

```
"security": {
 "access": {
 "requiresAuthentication": false
 }
 }
```

Conversely, the service-request-list flow requires the user to log in and have the "User" application role, and has the following security metadata:

```
 "security": {
 "access": {
 "requiresAuthentication": true,
 "roles": ["User"]
 }
 }
```

How do I use custom job roles in DCS?

You can create a custom job role for Customer Self-Service users. It lets you create a custom object and assign privileges to that object.

Note: This step is optional for advanced setup of your application.

There are many reasons why you might create a custom job role for Customer Self-Service users. One possible reason is when you create a custom object and want to assign privileges to that object.

Follow these steps to assign custom job roles for use in Digital Customer Service:

1. Create a custom role by making a copy of one of the following predefined roles:
    - Customer Self-Service User
    - Customer Self-Service Account Administrator

2. Set the value of the appropriate Digital Customer Service profile option to the name of the custom role you created. The following profile options can be set to the name of the custom role:
    - `SVC_CSS_USER_ROLE_COMMON_NAME` If the value of this profile option is set to the name of a custom role, then all Customer Self-Service Users will be assigned this role when they're provisioned.
    - `SVC_CSS_ACCT_ADMIN_ROLE_COMMON_NAME` If the value of this profile option is set to the name of a custom role, then all self-service users who are assigned the Customer Self-Service Account Administrator role will receive this custom role.

    For more information about setting profile options, see *How do I set profile options for Self-Service Optimization?*.

    > **Note:** If you update these profile options to specify a custom role once your Digital Customer Service application is in use, you must perform a mass update of any existing users from the old roles to the new roles.

For each custom job role you create, you also must create a group in Identity Cloud Service. The external ID of the IDCS group you create must match the code of the Fusion Service custom job role that you created. For more information about creating IDCS groups, see *How do I set profile options for Self-Service Optimization?*.

Application Configuration Settings

Digital Customer Service application settings are configured using application level variables in the Digital Customer Service web application. This topic describes how to configure settings, followed by descriptions of the application variables.

Modify the Application Variables

This topic describes how to modify application variables for your Digital Customer Service application in the Oracle Visual Cloud Builder Service. The variables that are available for configuration are described in the topics that follow.

To modify the application variables:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Web Applications** tile.
4. Click `dcs`.

    > **Note:** This is the default name when using the Digital Customer Service Reference Implementation. If you have changed the name in Oracle Visual Builder, click on the modified name.

    The `dcs` tab appears.
5. Click the **(x)** icon (Variables).
6. Click a variable, and make your modifications.

Linking

This topic describes application variable related to Service Request and Knowledge Management article linking.

All of the application variables related to Service Request and Knowledge Management article linking are contained in the **linkConfig** application variable. When you use the Digital Customer Service Reference Implementation template, the article linking section is configured as follows:

```
{
 "SERVICE_REQUEST_CRM": {
 "pattern": "\\b(SR|Bug)(\\d{10})\\b"
 },
 "KNOWLEDGE_LINK": {
 "pattern": "\\b((?:SOL|FAQ)\\d+)\\b",
 "keyProperty": "IMDocumentId"
 },
 "SERVICE_REQUEST_HCM": {
 "pattern": "\\b(Ticket)(\\d{10})\\b"
 }
}
```

The **SERVICE_REQUEST_CRM**, and **SERVICE_REQUEST_HCM** object type lines control the Service Request linking. The **KNOWLEDGE_LINK** object type line controls the Knowledge Management article linking.

By default in the Digital Customer Service Reference Implementation template, **SR** and **Bug** are defined as the case insensitive prefix that identify CRM Service Requests. Similarly, **Ticket** is defined as the case-insensitive prefix that identifies HCM Service Requests. For Knowledge Management articles, **SOL** and **FAQ** can be used interchangeably as the case-insensitive prefix that identifies articles. These prefixes are defined in the **pattern** application variable in each object type.

For example, if you want to change the prefix for Service Requests specifically to include both SR and Service Request as the prefix patterns, modify the **SERVICE_REQUEST** object type as follows:

```
"SERVICE_REQUEST_CRM": { "pattern": "\\b((?:SR|Service Request)\\d{10})\\b"},
```

Configure Product and Category Filtering

This topic describes the application variables related to product and category filtering.

- **dcsCategoriesOnly** Specifies the categories to display in your Digital Customer Service application. When set to **true**, categories with the **CSSFlag** set to **false** in the REST API won't be displayed in the **Category Selector**.
- **dcsProductsOnly** Specifies the products to display in your Digital Customer Service application. When set to `TRUE`, only products with `Enable for Customer Self Service` set to `YES` will be displayed in the **Product Selector**. When set to `FALSE`, all products in the Oracle Fusion Service product catalog are displayed.

Configure Product and Category Recent Selections

This topic describes the application variables related to product and category recent selections.

- **recent-selections** Contains the recent selections. This variable should be mapped to an application level variable. Up to five recent selections are displayed, with the most recent listed first.
- **recent-list-limit** Determines the number of rows that are displayed.
- **show-recent** Determines if recent selections are displayed.

Configure Language Defaults in Knowledge Management

This topic describes the application variables related to language defaults in Knowledge Management.

This configuration is required when more than one region is supported for the same language, because the default from the configuration file is used. Also for the API that retrieves the Knowledge Management locales, a `localeId` must be provided in the `kmauthtoken`, so that value is taken from the configuration file.

Oracle Knowledge Management supports a predetermined set of locales for knowledge searches. If the **Accept-Language** header element of the knowledge search REST request doesn't match one of the supported locales, an HTTP 400 error is returned. To prevent this, the locale specified in the **Accept-Language** header is overridden. This logic that determines the override locale is as follows:

1. Query the Server for all the supported locales and cache it.
2. Get the locale preference from the Oracle JET locale configuration using oj.Config.getLocale();

   > **Note:** Oracle JET determines the locale for locale-sensitive operations in the following order: locale specification in the RequireJS configuration, lang attribute of the HTML tag, navigator.language browser property or navigator.userLanguage Internet Explorer property.

   - If the locale preference is on the supported locale list, use it.
   - If the locale preference isn't a supported locale, extract the language from it.
3. Search the supported locale list by language. If only one match is found, use it.
4. Search `kmLanguageDefaults` in the `kmConfig` application variable. If a match is found, use it.
5. Pick the default locale for any language not on the list and use it.

The `kmLanguageDefaults` variable is the child of `kmConfig` variable. It maintains a mapping of language to locale.

The Interface ID for Knowledge Requests

This topic describes the application variables related to Interface ID defaults for Knowledge Management articles.

Use the `kmInterfaceId` variable , which is a child of the `kmConfig` top level application variable, to control what type of Knowledge Management articles appear in your Digital Customer Service application.

kmInterfaceId Variable Value	Effect
`1`	Only Oracle Fusion Service articles appear.
`2`	Only Oracle HCM Cloud articles appear.
`-1`	Both Oracle Fusion Service and Oracle HCM Cloud articles appear.

Application Configuration Settings

The Digital Customer Service application determines whether or not the user registration request requires an account key based on the value of the application variable userRegistrationType. This variable can have one of two values:

- Contact. Users registering as self-service users must provide an account key. When the user registration request is approved, the contact representing the user is associated with a specific account.
- Consumer. Users registering as self-service users do not have to provide an account key. When the user registration request is approved, the contact representing the user is created as a standalone contact.

The default value is contact.

To set the value of the userRegistrationType variable:

1. Open your Digital Customer Service application in the Visual Builder designer.
2. Select **Web Applications**.
3. Select **dcs**.
4. Select **Variables and Types**.
5. Select **userRegistrationType**.
6. Change the default value to either **contact** or **consumer** based on the requirements of your application.

Apply Themes to your Digital Customer Service Application

This topic describes how to apply themes to your own Digital Customer Service application.

Oracle JET includes themes that provide styling across a web or hybrid mobile application. You can use these themes as provided, or you can configure them manually and through the tooling.

For more information about applying themes to your Digital Customer Service application, refer to the chapter related to applying themes in the Developing Applications with Oracle JET guide, in the Related Topics.

Related Topics

- Developing Applications with Oracle Visual Builder
- Developing Applications with Oracle JET

Change the Appearance of your Digital Customer Service Application

This topic describes how to change the appearance of your own Digital Customer Service application.

The Oracle Digital Customer Service Reference Implementation template has been styled to enhance its appearance. This has been done by changing objects and adding styles to the `app.css` file.

To change objects in the `app.css` file:

1. Navigate to the Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click **Web Apps**.
4. Expand `dcs`, expand `resources`, expand `css`, then click `app.css`.

   The `app.css` tab appears.
5. Find and configure the object that you want to change. For example, to set the header of the Reference Implementation template to a transparent black, change the **odcs-header** CSS class selector:

```
 .odcs-header {
 background-color: rgba(0, 0, 0, 0.7);
 height: 58px;
}
```

CSS Classes defined in `app.css` can then be referenced in the HTML of the application. For example we've this code in `pages/shell-page.html`:

```
<header role="banner" id="header" class="odcs-header oj-web-applayout-header>"
```

You can also use Oracle JET themes to provide consistent appearance of components across your Digital Customer Service application. For more information about Oracle JET themes see the `Theming Applications` chapter in the Developing Applications with Oracle JET guide, in the Related Topics.

Once an Oracle JET theme is created it can be added uploaded to `resources/css` and then referenced in the `index.html` of the application with code like this:

```
...
 <link type="text/css" rel="stylesheet" href="resources/css/app.css">
 <link type="text/css" rel="stylesheet" href="resources/css/myJETTheme.css">
...
```

Related Topics

- Theming Applications

Custom Objects

If you create custom objects for either Fusion Sales or Fusion Service, you must use Application Composer. If you use the Business Objects feature in Visual Builder your work won't be backed by disaster recovery operations. Only use the Business Objects feature in Visual Builder for transient data.

Related Topics

- Configuring Applications Using Application Composer

6 Additional Feature Configuration

How do I configure multiple business units with Digital Customer Service?

When you have multiple business units, additional configuration is required once you have created your Digital Customer Service applications. Each business unit must have its own Digital Customer Service application.

Note: Only one business unit is supported per Digital Customer Service application.

Once you have created your Digital Customer Service applications, follow the instructions in the following sections of this topic:

1. Locate the business unit ID in Oracle Fusion Service.
2. Specify the business unit ID in the Digital Customer Service application.
3. Configure the Business Unit ID for the Open Service Requests list.

Locate the Business Unit ID

To locate the business unit ID for your Digital Customer Service application:

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. In the **Setup and Maintenance** work area, go to the following:
    - Offering: Service
    - Functional Area: Company Profile
    - Task: Manage Business Unit
3. Locate your business unit in the **Search Results** list and copy the value in the **BusinessUnitId** column.
   If you don't see a **BusinessUnitId** column, click the **View** menu to access the **Columns** menu, and then select the columns to display.

   Note: You will need to use the value that you copied in "Specifying the Business Unit ID and Product Catalog Usage Code in the Digital Customer Service Application" later in this answer.

Locate the Product Catalog Usage Code

To locate the Product Catalog Usage Code for your Digital Customer Service application:

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. 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

3. Copy the value in the **Business Unit Profile Value** text box.

   You will need to use the value that you copied in "Specifying the Business Unit ID and Product Catalog Usage Code in the Digital Customer Service Application" later in this answer.

   > **Note:** If you haven't yet set the scope for tasks, the **Select Scope** dialog box appears.

Specify the Business Unit ID and Product Catalog Usage Code in the Digital Customer Service Application

Once you have located the business unit ID and product catalog usage code in Oracle Fusion Service, you must specify them in your Digital Customer Service application.

To specify the business unit ID and product catalog usage code:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Web Apps** tile.
4. In the **Web Apps** tree, click `dcs`.

   A `dcs` tab appears.
5. Click the **(x)** (Variables) icon.
6. Set the business unit ID:
   a. Click **businessUnitId**.
   b. In the **Default Value** text box, specify the value that you copied in Step 3 of the Locating the Business Unit ID task.
7. (Optional) Set the non-default usage code:
   a. Click **usageCode**.
   b. In the **Default Value** text box, specify the value that you copied in Step 5 of the Locating the Product Catalog Usage Code task.
8. Refresh your Digital Customer Service application.

> **Note:** When adding the **Chat** or **Category Selector** to a page, you must ensure that your **businessUnitId** property is bound to `$application.variables.businessUnitId`. Moreover, when adding the **Product Selector** to a page, its **usageCode** property must be bound to `$application.variables.usageCode`.

Configure Installed Base Asset Components

How do I configure Installed Base Assets?

Here's the configuration steps you must perform to make Fusion products manageable as Installed Base Assets.

The Digital Customer Service product picker, by default, shows all products flagged as Enabled for Customer Self Service. You must set an additional property in Fusion Service to enable a product to be managed as an Installed Base Asset.

1. Sign in to Fusion Service as an administrator or setup user.
2. From the home page, select **Product Management,** then click the **Product Information Management** tile.
3. Click the **Tasks** drawer icon, and then click the **Manage Items** link.
   The Manage Items form appears.
4. Perform a search for your product, then, in the search results, select the link for the product.
5. In the product detail page, select the **Specifications** tab, then from the **Item Organization** list, select **Service**.
6. Update the products to make them manageable as Installed Base Assets, locate the **Assets** area of the view, and do one of the following:
   a. Click the Enable Asset Tracking drop down list and select: **Full Lifecycle**.
   b. Or, click the **Enable Asset Maintenance** drop down list, and select **Yes**.

Configure Create and Update Function Security Policies for Installed Base Asset Components

The seeded Customer Self-Service Proxy User role has privileges to view Installed Base Assets but doesn't have Update or Create privileges by default. You must add these privileges using Security Console.

For more information on enabling installed base assets, see the Related Topics link.

Create a New Role

The Customer Self-Service Proxy User role has privileges to view Installed Base Assets but doesn't have update or create privileges by default. You must add these privileges using Security Console.

1. Using Navigator, expand the **Tools** section, then select **Security Console**.
2. Click **Create Role**.
3. Complete the fields as shown in the following table:

Field	Value
Role Name	Create Update Assets
Role Code	Create_Update_Assets

4. Click **Next.**
5. Click **Add Function Security Policy**.
6. In the Add Function Security Policy pop up window, do the following:
   a. In the Search field, enter: **Create Customer Assets by Service**.
   b. Click **Add Privilege to Role**.
   c. In the Search field, enter: **Update Customer Assets by Service**.
   d. Click **Add Privilege to Role**.

e. Click Cancel to close the window.
7. Click the **Users** train stop.
8. Click **Add User**.
9. In the Add User pop up window, do the following:
    a. In the Search field, enter: **PUDS_CSS_USER**.
    b. Click **Add User to Role**.
    c. Click Cancel to close the window.
10. Click **Next**.
11. Review the summary information, then click **Save and Close**.

Disable Asset Management

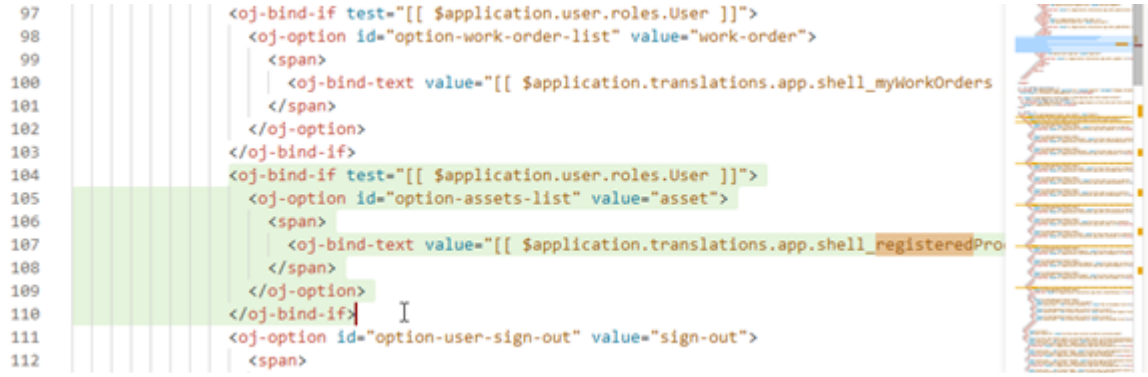
You can hide the asset registration link from the application User menu if you don't want this functionality to be displayed to your users.

You can also completely remove the asset-related flows from Visual Builder if you want to remove this functionality in your application altogether. Here's how you do both.

Remove the Registered Products Link

Here's how you delete asset registration from the User menu in the Digital Customer Service Application.

1. In Visual Builder, open your application, then click **Web Apps**.
2. Expand **dcs**, and then expand **Root Pages**.
3. Click **shell**.
4. In the design palate, click the **Code** button.
5. Search for "registered" and select the code entry as shown in the following example:



6. Delete the lines of code.

Delete the Asset Flows

If you want the asset flows removed entirely from your application, here's how you do it:

1. In Visual Builder, open your application, then click **Web Apps**.
2. Expand **dcs**, and then expand **Flows**.
3. Right-click the following three asset related flows and choose **Delete**.

- asset-detail
- asset-list
- asset-register

All asset related functionality will now be removed from your Digital Customer Service application.

Integrate Chat in Your Digital Customer Service Application

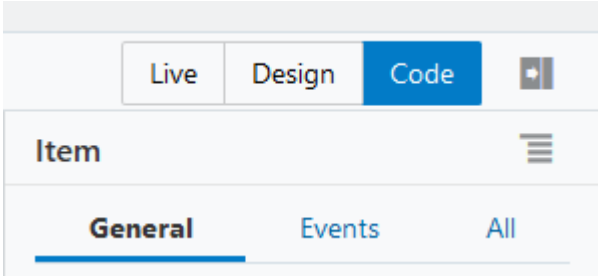
In the Digital Customer Service reference implementation, the chat component is deployed in the footer of the shell page so that chat sessions persist across page navigation.

If you want to create another shell page or show the chat dialog from an action, follow these steps.

Create a New Root Page

In the Digital Customer Service reference implementation, the chat component is deployed in the footer of the shell page so that chat sessions persist across page navigation. If you want to create another shell page follow these steps:

1. In Visual Builder, expand the **Root Pages** node, open the original shell page where the chat component appears.
2. Click the **Code** button:



3. Copy the oj-odcs-chat component from the footer of the original shell page, and then replace the oj-odcs-chat element in the foot of the new shell page. In the Components Search field, enter **chat**.

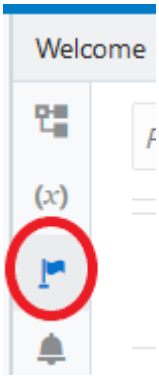
   Copying the code from the original shell page ensures that you have the correct ID, translations and correct references to application variables.

Show Chat with an Action

If you want to show the chat dialog from an action, follow these steps.

1. In the Web Apps navigator, click your application icon.

2. Click the **Actions** icon:



3. In the list of Actions, locate and select the **ShowChatAction** action chain.
4. In the Action Chain properties window, make sure the following entries are correct:
    - Component: `{{ document.getElementById('odcs-chat') }}`
    - Method Name: open.

Configure Chat to be Offered Only When Agents are Available

You can configure chat so that it's offered as an option to customers only when an agent is available. You just need to do the following two steps:

1. Set SVC_CHAT_INLAYS_ACCESS_ENABLED profile option to **Yes**.
2. Set the default value of the DCS `chatPollingEnabled` application variable to **True**.

Disable the Chat Download Button

If you don't want your users to have the ability to download chat transcripts at the end of a chat session, you can hide the Download button by adding to the application's app.css file:

```
div[slot="endChat"] > div > div:nth-child(2) > div:nth-child(2) {
    visibility: hidden;
}
```

Map Roles for Digital Customer Service

Now you map roles to let the Oracle Visual Builder role to match with Oracle Fusion Service. Once you have done this task you will be able to manage user roles in the Oracle Fusion Service and your Digital Customer Service application similarly.

You must follow the process for mapping roles in this topic if you want to define additional roles.

Map a New Role

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.

3. Click the **Menu** icon and select **Settings**.
4. Click the **User Roles** tab.
5. Click **Add Role**.
6. Enter a role name in the **Role** text box.
7. In the **Mapping** list, select the role to which to map.
8. Click the **Check Mark** icon to complete the mapping.

Edit an Existing Role Mapping

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Menu** icon and select **Settings**.
4. Click the **User Roles** tab.
5. Hover over the role that you want to edit in the **User Roles** list, then click the **Pencil** icon.
6. Edit the role name in the **Role** text box.
7. In the **Mapping** list, edit the role to which to map.
8. Click the **Check Mark** icon to complete the mapping.

Remove an Existing Role Mapping

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Menu** icon and select **Settings**.
4. Click the **User Roles** tab.
5. Hover over the role that you want to remove in the **User Roles** list, then click the **Trash Can** icon.

   The user role is removed.
6. Click **Close**.

Set Up Products, Categories and Knowledge Articles

Here are instructions on how to create products, categories, and author knowledge articles specific to Digital Customer Service.

Product items and groups are used within the Digital Customer Service application to provide better support and knowledge article navigation. Also, product items and groups help you associate service requests for Oracle Fusion Service process flows. So, when you're creating and using product items and groups, think of them as another way to improve your customer's experience.

To configure products and categories for service request management:

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Configure the products that are available in service requests.

> **Note:** When creating your products, the following options must be selected: **Eligible to Sell**, **Eligible for Service** and **Enable Customer Self-Service**.

3. Configure the available categories for service requests.
4. Author Knowledge articles that you want users to have access to in their Digital Customer Service applications.

Create a Digital Customer Service Icon in Oracle B2B Service

Sometimes, when users perform certain actions in the Digital Customer Service application, they're redirected to the Oracle Fusion Service Dashboard. To make this path easier for your users, Oracle recommends that you include a Digital Customer Service icon within the user's dashboard.

> **Note:** If you use a third-party identity management solution with Oracle Fusion Service, you must address the redirect issues within your deployment.

Follow the procedures in this topic in the listed order to create a Digital Customer Service icon in Oracle Fusion Service:

1. Create a New Sandbox.
2. Create a New Card Icon for Accessing Digital Customer Service.
3. Validate the Card Icon.
4. Publish the Sandbox.

Create a New Sandbox

To create a new sandbox:

1. Sign in to Oracle Fusion Service.
2. Click **Navigator > Configuration > Sandboxes**.
3. Click the **+** icon (New) to create the new sandbox.
4. Select the sandbox created in Step 3.
5. Click **Set as Active**.

Create a New Card Icon for Accessing Digital Customer Service

To create a new card icon for accessing Digital Customer Service:

1. Sign in to Oracle Fusion Service.

   > **Note:** Ensure that your sandbox is set as active.

2. Navigate to the **Configuration** work area and click **Structure**.
3. From the Create menu, select **Create Page Entry**.
4. Enter information for the new card:
   - **Name**. Specify **DCS**.
   - **Icon**. Select an icon.
   - **Group**. Select **Service**.
   - **Show on Navigator**. Select **EL Expression**, then select the **Edit** and paste the following expression:

```
#{!securityContext.userInRole['ORA_PER_EMPLOYEE_ABSTRACT']}
```

- **Show on Springboard**. Select **EL Expression**, then select the **Edit** and paste the following expression:

```
#{!securityContext.userInRole['ORA_PER_EMPLOYEE_ABSTRACT']}
```

- **Link Type**. Select **Static URL**.
- **Destination**. Specify the URL for your Digital Customer Service web application.

5. Click **Save and Close**.

Validate the Card Icon

To validate the card icon:

1. Sign in to Oracle Fusion Service.
2. Click **Navigator > Configuration > Sandboxes**.
3. Select the sandbox created in the Create a New Sandbox topic.
4. Click **Set as Active**.
5. Navigate to the **Home** page.

   The **DCS** icon appears on the **Home** page and in the **Navigator**.

Publish the Sandbox

To publish the sandbox:

1. Sign in to Oracle Fusion Service as a user with the Sales Administrator job role.
2. Click **Navigator > Configuration > Sandboxes**.
3. Select the name of the sandbox created in the Create a New Sandbox topic.

   The Sandbox Details dialog appears.
4. Click **Publish**.

Configure Proactive Knowledge

This topic provides an overview of the proactive knowledge feature and describes how to configure variables relating to proactive knowledge.

Overview of Proactive Knowledge

The proactive knowledge feature enables you to display relevant knowledge articles to users prior to creating a Service Request (SR). Having access to knowledge articles helps customers self-service their own issues, possibly avoiding creating SRs altogether.

When the user attempts to submit an SR, the application uses the title of the SR to search the knowledge base. If the knowledge search returns matches, the application shows a list of relevant knowledge articles to the user. If the user determines that one of the articles answers their question, then the SR isn't submitted. If the knowledge articles don't help, the user can go ahead and create the SR. If the knowledge search doesn't return any matches, the application automatically submits the SR.

You can control if proactive knowledge is displayed at all, or based on how many knowledge searches a user has completed, or even by the number of articles that the user views. Also, you can configure the number knowledge articles that are displayed.

Control when Proactive Knowledge is Displayed

Proactive knowledge display options are controlled using variables. This table lists and describes the variables, and also contains an explanation of the accepted values that can be assigned to each variable

Variable	Description	Accepted Values

`isProactiveKnowledgeOn`	Controls if proactive knowledge is displayed during SR creation.
	
	Note:
	If this variable is set to `false`, proactive knowledge is entirely disabled. Consequently, no other variables related to proactive knowledge need to be configured.
		`true`
		
		or
		
		`false`
		
		The default value is `true`.

`disableProactiveKnowledgeSearchCount`	The number of knowledge searches that a user must perform to prevent proactive knowledge articles from being displayed. For example, if this value is set to 2, then if a user performs at least two searches for knowledge articles, then proactive knowledge isn't displayed.	Positive integer The default value is `1`.
`disableProactiveKnowledgeViewCount`	The number of knowledge articles that a user must view to prevent proactive knowledge articles from being displayed. For example, if this value is set to 3, then if a user views at least three knowledge articles, then proactive knowledge isn't displayed.	Positive integer The default value is `2`.

To configure the variables that control if and when proactive knowledge is displayed:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Web Applications** tile.
4. In the **Web Apps** tree, expand `dcs`, and then expand **flows**.
5. Click on **service-request-create**.

   The **service-request-create** tab opens.
6. Click the **(x)** (Variables &Types) icon.
7. Modify the **Default Value** field of one or more variables of the following variables:
   - `isProactiveKnowledgeOn`
   - `disableProactiveKnowledgeSearchCount`

- `disableProactiveKnowledgeViewCount`

Note: If the number of knowledge searches performed is equal to or greater than the value in `disableProactiveKnowledgeSearchCount` or the number of knowledge articles viewed is equal to or greater than the value in `disableProactiveKnowledgeViewCount`, then proactive knowledge isn't displayed.

Control How Many Knowledge Articles are Displayed

You can control the maximum number of knowledge articles to display when proactive knowledge displayed. The variable that controls this is `kmArticlesPageSize`.

Note: The default value assigned to `kmArticlesPageSize` is `5`.

To configure how many knowledge articles to display:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Web Applications** tile.
4. In the **Web Apps** tree, expand `dcs`, then expand **flows**, and then expand **service-request-create**.
5. Click on **service-request-create-answers**.
   The **service-request-create-answers** tab opens.
6. Click the **(x)** (Variables &Types) icon.
7. Click **kmArticlesPageSize**.
8. Enter the number of articles to display in the **Default Value** field.

How do I set up Digital Customer Service self-service registration?

There are two types of Self-Service registration requests, Fusion and B2C.

- You allow Fusion requests by setting the value of the SVC_CSS_ALLOW_CONTACT profile option to TRUE.
- You allow B2C requests by setting the SVC_CSS_ALLOW_CONSUMER profile option to TRUE.

The following users can submit self-service registration requests::

- Anonymous Users
- User with Customer Self-Service Administration duty role
- User authenticated by IDCS through the Proxy User Data Service

Here's how the self-service registration request works:

First, the registration request is validated. If no errors are found, the status of the request is set to ORA_CSS_PENDING. If the value of profile option (either SVC_CSS_ALLOW_CONTACT for Fusion, or SVC_CSS_ALLOW_CONSUMER for B2C) is set to TRUE then the process continues.

If the profile option is set to FALSE, then the request must be manually initiated by a user with either the Customer Self-Service Account Administrator job role or Customer Self-Service Account Administration

duty role. The name of the profile option is SVC_CSS_SELF_REG_AUTO_APPROVE for Fusion requests and ORA_SVC_CSS_SELF_REG_B2C_AUTO_APPROVE for B2C requests.

Here's the expected result:

- The user account in IDCS with the Customer Self-Service User job role (or job role given by profile option SVC_CSS_USER_ROLE_COMMON_NAME) will be present.
- The contact in Fusion Service, stamped with the GUID of the user account in IDCS.
- The contact is given the Self-Service User role.
- If the request was of the Fusion type:
    - A relationship between the business account and the contact is indicated.
    - If the user is the first user of a business account the user is given the Customer Self-Service Account Administrator job role (or a job role given by the SVC_CSS_ACCT_ADMIN_ROLE_COMMON_NAME profile option).

The primary attributes that influence the Self-Service registration are:

- Request Type Code (RequestTypeCd). If this attribute is absent in the request payload or if the value of this attribute is ORA_CSS_REQ_TYPE_CONTACT then its a Fusion request. If the value of this attribute is ORA_CSS_REQ_TYPE_CONSUMER then it's a B2C request.
- Account Key (AccountKey). For Fusion requests, this attribute is expected to identify a unique business account. The profile option SVC_CSS_ACCT_KEY_FIELD determines the attribute of the Account object whose value must be the specified account key value. The default value of this profile option is OrganizationName but it can be set to any attribute of the account object whose value is unique to a single account.
- Email Address (EmailAddress). This attribute is the default value of the SVC_CSS_REG_FLD_CONTACT profile option and is used to locate an existing contact. You can set this profile option to any attribute on the Self-Service Registration object. The corresponding attribute on the Contact object is located either using auto mapping logic or using the value of the SVC_CSS_REG_CONT_MAP profile option. The email address of the located contact and the email address of the registration request must be the same.
- Login ID (LoginId). If a value for this attribute isn't specified, the value will be set to the value of the attribute identified by the SVC_CSS_SIGN_IN_ATTR_NAME profile option whose default value is EmailAddress. Login ID is used to locate a user account in the Identity database.

The following table shows the actions taken based on the result of searching for existing contact and user account:

Search Result	Action
Contact doesn't exist but the user account exists.	A contact record is created and the GUID of the user account is stamped on the contact if the user submitting the registration request is authenticated by IDCS or the user submitting the registration request has been given the Associate User With Contact privilege. The Associate User With Contact privilege is given by default only to users with the Customer Self-Service Administration duty role.
Contact exists and the user exists and contact record isn't stamped with the GUID of the user account.	The GUID of the user is stamped on the contact provided the user submitting the registration request is authenticated by IDCS and the email address of the user account is same as that of the registration request or the user submitting the registration request has been given the Associate User With Contact privilege.
Contact exists and the user exists and contact record is stamped with the GUID of the user account.	If this is a Fusion request, the user already has a self-service user role for one account and is granted the self-service role for another account.

Search Result	Action
Contact exists but user account isn't found.	A user account is created and the contact is stamped with the GUID of the user account.
Neither a contact nor a user account is found	The records are created and the contact record is stamped with GUID of the user account.

After the contact record is created, the attributes of the Self-Service Registration object are copied over to the Contact object. If the name of an attribute of the SelfRegistration object (ignoring _c) is same as that of the Contact object then the value of that attribute is copied over. If the names aren't the same then the SVC_CSS_REG_CONT_MAP profile option can be used to map an attribute of the SelfRegistration object to an attribute of the Contact object.

How do I configure the Self-Service Registration object in Digital Customer Service?

Use this topic to configure the Self-Service Registration object.

Overview of the Self-Service Registration Object

Digital Customer Service self-service registration requests are submitted using the Self-Service Registration object. This object is extensible and can be configured using the Application Composer in Oracle Fusion Service. With the Application Composer, you can add new fields, validation rules and triggers to the object.

The payload of the REST request to the Self-Service registration resource can supply values for the following attributes:

Basic attributes:

Attribute Name for the SelfRegistration Object	Display Name	Attribute in the Contact Object	Type
EmailAddress	Email Address	EmailAddress	Text
AccountKey	Account Key	AccountKey	Text
FirstName	First Name	FirstName	Text
MiddleName	Middle Name	MiddleName	Text
LastName	Last Name	LastName	Text
PlaceOfBirth	Place Of Birth	PlaceOfBirth	Text

Address attributes:

Attribute Name for the SelfRegistration Object	Display Name	Attribute in the Contact Object	Type
PrimaryAddressLine1	Primary Address Line 1	Address1	Text
PrimaryAddressLine2	Primary Address Line 2	Address2	Text
PrimaryCountry	Primary Country	Country	LOV
PrimaryCity	Primary City	City	LOV
PrimaryState	Primary State	State	LOV
PrimaryProvince	Primary Province	Province	Text
PrimaryPostalCode	Primary Postal Code	PostalCode	LOV
PrimaryAddressType	Primary Address Type	AddressType	Text
SecondaryAddressLine1	Secondary Address Line 1	Address1	Text
SecondaryAddressLine2	Secondary Address Line 2	Address2	Text
SecondaryCountry	Secondary Country	Country	LOV
SecondaryCity	Secondary City	City	LOV
SecondaryState	Secondary State	State	LOV
SecondaryProvince	Secondary Province	Province	Text
SecondaryPostalCode	Secondary Postal Code	PostalCode	LOV
SecondaryAddressType	Secondary Address Type	AddressType	Text

Contact Point attributes:

Attribute Name for the SelfRegistration Object	Display Name	Attribute in the Contact Object	Type
RawWorkPhoneNumber	Raw Work Phone Number	RawPhoneNumber	Text

Attribute Name for the SelfRegistration Object	Display Name	Attribute in the Contact Object	Type
RawMobileNumber	Raw Mobile Number	RawPhoneNumber	Text
RawHomePhoneNumber	Raw Home Phone Number	RawPhoneNumber	Text
WorkPhoneCountryCode	Work Phone Country Code	PhoneCountryCode	Text
MobileCountryCode	Mobile Country Code	PhoneCountryCode	Text
HomePhoneCountryCode	Home Phone Country Code	PhoneCountryCode	Text
WorkPhoneAreaCode	Work Phone Area Code	PhoneAreaCode	Text
MobileAreaCode	Mobile Area Code	PhoneAreaCode	Text
HomePhoneAreaCode	Home Phone Area Code	PhoneAreaCode	Text
WorkPhoneExtension	Work Phone Extension	PhoneExtension	Text
MobileExtension	Mobile Extension	PhoneExtension	Text
HomePhoneExtension	Home Phone Extension	PhoneExtension	Text
WorkPhoneNumber	Work Phone Number	PhoneNumber	Text
MobileNumber	Mobile Number	PhoneNumber	Text
HomePhoneNumber	Home Phone Number	PhoneNumber	Text

During the approval process, a contact record is created, and the attributes of the Self-Service Registration object can be transferred to the Contact object. The value assigned to the `SVC_CSS_REG_CONT_MAP` profile option determines which attributes in the Self-Service Registration object are transferred to which attributes in the Contact object. The default is an empty string. You specify a value for this profile option only if the name of the attribute in the Self Registration object is different from the name in the Contact object. Cases where they may happen are if you have created a custom attribute for an object. Custom attributes are designated with an _c, such as PlaceOfBirth_c. For this use case, you ignore the _c when determining whether an attribute maps or not. So, let's take the custom attribute in the Self Registration object PlaceOfBirth_c. Since the Contact object has a out of the box attribute called PlaceOfBirth no mapping is required since the two values match. If, however, the name of the custom attribute was BirthPlace_c the value of this profile option would then be BirthPlace_c:PlaceOfBirth.

Usage Example

For business reasons, if additional information needs to be gathered about the user submitting a registration request, custom fields can be added to the Self-Service Registration Object. If an additional field is a required, then a value must be provided in the REST request sent to the Self-Service Registration object.

If a new required custom attribute is added to the Contact object, a new custom attribute must also be added to the Self-Service Registration object and then specified in the `SVC_CSS_REG_CONT_MAP` profile option. This will transfer the value of the new attribute of the Self-Service Registration object to the new attribute of the Contact object.

For example, let's say there's a custom attribute in the Self Registration object called PlaceOfBirth_c. This attribute can be added to the Self-Service Registration object and mapped to the **Place of Birth** attribute that already exists on the Contact object. If the name of the attribute is `BirthPlace_c` then the value of `SVC_CSS_REG_CONT_MAP` profile option should be `BirthPlace_c:PlaceofBirth`. You can map multiple attributes using colon separating fields, and commas separating the pairs. For more information, see the entry in the Registration Profile Options table in *How do I configure DCS profile options?*.

> **Note:** The **API Name** of the new attribute is different from the **Name**.

Here are the tasks that you need to complete to address this use case:

1. Create the Field
2. Test the REST Request
3. Modify the Profile Option

Create the Field

First, you need to create the field.

To create the field:

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Create a sandbox for adding the Place of Birth field:
    a. Click **Navigator > Configuration > Sandboxes**.
    b. Click **Create Sandbox**.

    The **Create Sandbox** page appears.
    c. Enter a name in the **Name** field.
    d. From the All Tools list, select **Application Composer**.
    e. Click **Create**.
    f. In the **Available Sandboxes** list, click the name of the sandbox name that you specified in step c.
    g. Click **Enter Sandbox**.
3. Navigate to **Application Composer**.
4. Expand **Objects**, then **Standard Objects**, then **Self-Service Registration**, and then click **Fields**.

    The **Fields** page appears.
5. Click **Create a custom field**.
6. Click the **Text** option, then click **OK**.
7. Specify the following for the Date field options:
    - In the **Display Label** field, enter the following string:

`Birth Place`

- The **Name** field will be pre-populated based on the name that you entered for the **Display Label**, without any spaces.
- The **API Name** field will be pre-populated based on the name that you entered for the **Display Label**, without any spaces, and typically with the following suffix: `_c`

> **Tip:** Note the value assigned to the **Birth Place** field, because it will be assigned to the `SVC_CSS_REG_CONT_MAP` profile option in the Modify the Profile Option task, later in this topic.

- Deselect the **Required** option in the **Constraints** section.
- Select the **Updatable** option in the **Constraints** section.
- Deselect the **Searchable** option in the **Constraints** section.
- Select the **Include in Service Payload** option in the **Constraints** section.

8. Click **Save and Close**.

Test the REST Request

Use the following sample curl command to test the REST request. You must set the profile option before you complete your testing. Also, given a meaningful value for the Birth Place (`Birthplace_c`) field such as "2001-01-01".

> **Note:** This is only an example. Your curl command must include details relevant to your deployment.

```
curl -X POST \
 https://myhost.us.example.com/crmRestApi/resources/11.13.18.05/selfRegistrations \
 -H 'Accept: application/json' \
 -u "<user_name>:<password>" \
 -H 'Content-Type: application/vnd.oracle.adf.resourceitem+json' \
 -d '{
 "AccountKey": "HDFC Bank",
 "PersonFirstName": "Lilly",
 "PersonLastName": "Inigo",
 "EmailAddress": "lilly.inigo@example.com" ,
 "BirthPlace_c":"New York"
```

Modify the Profile Option

To modify the `SVC_CSS_REG_CONT_MAP` profile option so that it includes the Place of Birth field:

1. Sign in to Oracle Fusion Service as administrator or a setup user.
2. In the **Setup and Maintenance** work area, go to the following:
   - Offering: Service
   - Functional Area: Digital Customer Service
   - Task: Manage Digital Customer Service Profile Options
3. Click the `SVC_CSS_REG_CONT_MAP` profile option.
4. Add the following profile option value to the list of values:

   `BirthPlace_c:PlaceOfBirth`
5. Click **Save and Close**.

Usage Example with Multiple Mappings

Now, let's briefly consider a scenario with multiple mappings.

First, you specify the case sensitive name and value pairs to map the fields of the Registration View object to the Contact View object. Here's how you do it: in the following way:

- `reg_field1:contact_field1,reg_field2:contact_field2`

Where, the `reg_field1` is the PlaceOfBirth_c in the Registration View Object which is itself a custom object created in Application Composer.

The `contact_field` is the PlaceOfBirth field in the Contact View object. This attribute is already present in the Contact object.

So the mapping would be: `reg_field1:contact_field1 LIKE BirthPlace_c:PlaceOfBirth`

Set Up Visual Navigator

Visual Navigator, once configured, enables you to show a grid of tiles on a page to facilitate navigation within and outside of the Digital Customer Service application.

When you drop the component on a page in the DCS application, you must configure the component before anything appears in the UI. You must determine the number of grid tiles, the image and text that displays in each tile along with the action that's taken when each tile is clicked.

Find the Visual Navigator component in the list of Digital Customer Service components in the Components palette. Drag and drop the component from that location onto the page where you would like the Visual Navigator to appear. When you drop the component on the page, a NavigatorLoadAction action chain is created at the page level. You will need to modify this action to configure the items the navigator displays. A number suffix is added to the name if more than one navigator is added to a given page.

Specify Items for an Array

To configure the navigator items you must first specify the items that you want the navigator to show in an array. You can specify this either by an action chain variable or using a separate JSON file.

Specify Items for the Action Chain Variable

Now you specify the items as an action chain variable. You edit the NavigatorLoadAction and set the default value of the action chain variable items to be an array.

1. From the page flow that you dropped the Visual Navigator component on, select the Actions icon.
2. From the list of Actions, select **NavigatorLoadAction**.
3. Click the **Action Chains - Variables** tab.
4. Select the items array from the list of variables
5. In the **Default Value** field, enter your code.
   See the following example as reference.

```
[
 {
```

```
"title": "Happy Holidays",
"description": "Wishing all our customers a happy holiday season!",
"type": "info",
"icon": "fa fa-5x fa-glass-cheers"
},
{
"title": "Setup your Nimbus 3000",
"description": "How do I configure my Nimbus 3000?",
"type": "knowledge-article",
"id": "1003002",
"icon": "fa fa-5x fa-question"
},
{
"title": "My Service Requests",
"type": "service-request",
"operation": "list",
"icon": "fa fa-5x fa-ticket-alt"
},
{
"type": "product",
"id": "CRMITEM-AS54888-00182744",
"icon": "fa fa-5x fa-desktop"
},
{
"type": "my-profile",
"title": "My Profile",
"description": "View your account information.",
"icon": "fa fa-5x fa-user"
},
{
"type": "url",
"title": "Google",
"icon": "fa fa-5x fa-search",
"params": {
"url": "https://www.google.com"
}
}
]
```

Work With the JSON File

Create a JSON file on your local file system using the same format as the previous example. From the web application navigator in Visual Builder upload the JSON file in a folder under the application's resource folder. Do this by either dragging and dropping the file or by right-clicking the folder you want to upload the JSON file to and selecting Import from the menu.

If you need to edit the NavigatorLoadAction use these steps:

1. Add a Call Module Function action as the first action in the chain.
2. Set the Call Module Function to call the application level loadJSON function.
3. Set the path input parameter of the function to be the relative path of the JSON file under the resources directory.
4. Add an Assign Variables action after the newly added Call Module Function action.
5. Assign the result of the call module function to flow level items variable.

Navigation Items

The supported types and operations are defined in the NavigateAction on the shell page. By default, the following types and operations are supported but these can be extended by adding additional cases to the shell page's NavigateAction. If no operation is specified it defaults to details.

The supported types and operations are listed in the following table:

Description	Type	Operation	ID	Parameters	Additional Information
Static information	info				
Navigate to the Contact Us Page	contact-us				
Navigate to the Home page.	homepage				
Visit a knowledge article	knowledge-article		Knowledge article identifier. The kmContentId URL parameter when viewing the article in DCS.		
Navigate to the user's profile page	my-profile				
Navigate to the product page	product		Item number of product		The title of the navigation item will be the product's name loaded from Fusion Service.
Product Group	product-group		ProductGroupId of the product group		Title of the navigation item will be the product group's name loaded from Fusion Service.
Category	category		CategoryId of the category		Title of the navigation item will be the category's name loaded from Fusion Service.
Navigate to the service request create page.	service-request	create			
Navigate to the service request list page.	service-request	list			
Navigate to the work order list page	work-order	list			
Open a URL in a new tab	url			{	

Description	Type	Operation	ID	Parameters	Additional Information
				"url": "https://www.samplewebpage.c }	

Icons and Images

Items can have either an icon, such as a Font Awesome icon, or an image.

To apply an icon set the icon property of the item and specify the style classes required as its value, for example, fa fa-5x fa-user.

To use an image set the imageUrl property to the image's URL. Just a reminder, image names are case sensitive. Here's an example:

```
"imageUrl": "[[ $application.path + 'resources/images/service-request/SR0000038250.png' ]]"
```

Translations

To specify a key in the application resource bundle set the title or description of the item to the key prefixed with an exclamation mark, for example, !bundle_key.

Here's an example:

```
{
 "type": "my-profile",
 "title": "!homepage_visual_navigator_myProfile_title",
 "icon": "fas fa-4x fa-address-card odcs-home-visual-navigator-card-icon"
 }
```

Automatically Populate Key Information in Pre-Chat Form

You can automatically populate the product, category, and service request number in the pre-chat form, based on the page context in which the chat is initiated.

Here's how it works:

First, with the service request number on a service request form:

- If you're viewing a service request with a product (or product group) already selected and you initiate a chat, then the selected product (or product group) will be auto-populated in the pre-chat form.
- If you're viewing a service request with a category already selected and you initiate a chat, then the selected category will be auto-populated in the pre-chat form.
- If you're viewing a service request and you initiate a chat, then the service request number will be auto-populated in the pre-chat form.

Here's how fields are automatically populated on the Product Group landing page:

- If you're viewing a product landing page and you initiate a chat, then the selected product will be auto-populated in the pre-chat form.
- If you click through to a knowledge article from a product landing page, then the selected product will be auto-populated in the pre-chat form.

Now, the Product Group landing page:

- If you're viewing a product group landing page and you initiate a chat, then the selected product group will be auto-populated in the pre-chat form.

The Category Landing page:

- If you're viewing a category landing page and you initiate a chat, then the selected category will be auto-populated in the pre-chat form.
- If you click through to a knowledge article from a category landing page, then the selected category will be auto-populated in the pre-chat form.

And finally, here's the behavior with Knowledge search results:

- If you have done a knowledge search and selected a product from the product selector and you initiate a chat from the search results list or from a knowledge details page (from the search results), then the selected product will be auto-populated in the pre-chat form.
- If you have done a knowledge search and selected a category from the category selector and you initiate a chat from the search results list or from a knowledge details page (from the search results), then the selected category will be auto-populated in the pre-chat form.

The Chat component doesn't display the category, product or service request by default, so you must enable them by doing the following:

1. Open your Digital Customer Service application in Oracle Visual Builder.
2. Navigate to **Web Applications**, **dcs**, and expand **Root Pages**, and then click **shell**.
3. Click shell, and then click the **Page Structure** icon.
4. In the Page Structure list, search for **Chat**.
5. Click **Chat** to display the property inspector.
6. Within the chat attributes, locate the **category** property, and click the arrow to display the sub-category.
7. Change the **category.show** sub-category value to **true**.
8. Click the category arrow to close the sub-category.
9. In the property inspector, locate the product property in the Chat attributes, and click the arrow to display the sub-category.
10. Change the **product.show** sub-category value to true, then click the product property arrow to close to sub-category.
11. Now locate the service-request property in the Chat attributes property inspector and click the arrow to expand the sub-category.
12. Change the **service-request.show** sub-category value to true

How do I set up vanity URLs?

The Vanity URL feature enables you to map a custom domain to your Oracle Visual Builder instance and Digital Customer Service Application.

This topic describes how to provide a replacement domain to use instead of the typical Oracle domain. The Vanity URL feature is distinct from domain forwarding, where the browser gets redirected. With the CNAME mapping described here, your users will never see the Oracle domain in their browser.

To successfully set up a vanity URL, follow the following three tasks in this topic:

1. Configure a Domain Provider
2. Log a Service Request for Oracle Support
3. Set Up the Root URL App

Prerequisites

Before following the steps in this task you must make sure you own the sub-domain that you want to use and have access to the SSL certificate bundle information.

> **Note:** If you have multiple Digital Customer Service applications you must perform these configuration steps on each one.

Configure the Domain Provider

The following instructions represent a generic overview for configuring a domain provider. The exact steps might be different, depending on your domain provider.

1. Create or identify a subdomain to map to Oracle Visual Builder via a domain provider. For example: `https://myvanity.example.org`.
2. Add a CNAME record for the subdomain to map to the Oracle Visual Builder instance URL. For example, create a CNAME record for `https://myvanity.example.org` that points to `myvbinstance.builder.ocp.oraclecloud.com`.
3. Ensure a valid SSL certificate applies to that subdomain, either through your domain provider or through a valid certifying authority. For example, `Comodo`, `DigiCert`, or other.

   > **Note:** Even though it's possible to use a wildcard SSL certificate (`*.example.org`), the certificate bundle needs to be maintained on the server. Because of this, you will want to consider an SSL certificate, specifically for that subdomain. The following example shows a wildcard certificate issued for the `*.example.org` domain.

4. Ensure that you have extracted or exported the bundle that contains the certificate and private key, because the certificate bundle will be managed on the server.

   > **Note:** Depending on your domain provider, you may need to indicate that you want to use the certificate on your own server in order to download the bundle.

   You should now have a certificate along with a private key file.

Log a Service Request for Oracle Support

Now, you need to log a Service Request (SR) with Oracle Support to request your Oracle Visual Builder instance to be configured with your vanity domain.

1. Navigate to the Oracle Support site.
2. Sign in using your user name and password.
3. Create an SR. Your SR must include the following information:
   - The full VB URL Designer URL. For example: `myvbinstance.builder.ocp.oraclecloud.com`.

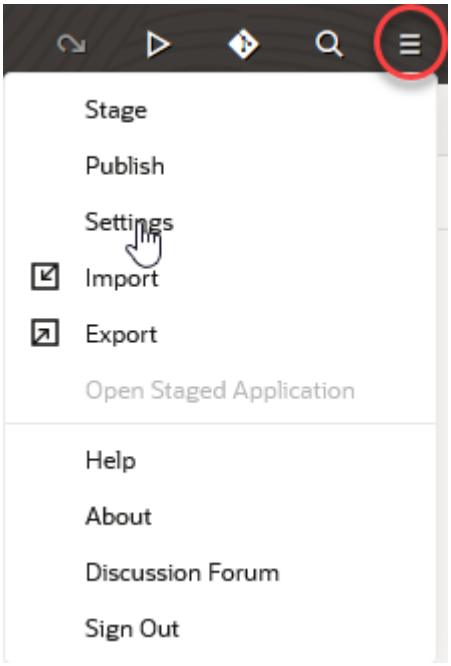
- This the CNAME as provided and the VB domain e.g. myvbinstance.builder.ocp.oraclecloud.com.
- The Certificate and Private key files. For example, the keys ending in `.cer` and `.key`.

Set Up the Root URL App

Follow the steps in this task to configure the custom URL for your domain.

To map a custom domain to your application:

1. In Visual Builder, click the **Web Applications** tab, and then click **Menu** and select **Settings**, as shown in the following example:



2. In the Settings dialog box, enter the URL into the **Vanity URL** text field and click **Close**.

   The URL must be the full URL that you want to use and it must use valid form (for example, https://myvanity.example.org).

After you publish your DCS application, a visitor can type the custom domain (for example, https://foo.example.org) in the browser to open the web application. The URL won't contain any additional path parameters because the app is loaded as the root domain.

Post-Configuration Verification Tasks

This topic describes the post-configuration verification tasks recommended after configuring your Digital Customer Service application. The tasks outlined in this topic apply only if you created your Digital Customer Service application using the Digital Customer Service Reference Implementation template.

Verify the Knowledge Search Component

To verify the Knowledge Search component:

1. Navigate to your Digital Customer Service application.
2. Enter the search text for the Knowledge Management articles that have been created.
3. Click the **Search** icon.
4. Verify that the search results match the search text.
5. Click the **Category** field to verify the list of categories are displayed.
6. Click the **Product** field to verify the list of products are displayed.
7. Sign in to your Digital Customer Service application as a self-registered user.
8. Enter the search text for the Knowledge Management articles that have been created.
9. Click the **Search** icon.
10. Verify that the search results match the search text.
11. Click the **Category** field to verify the list of categories are displayed.
12. Click the **Product** field to verify the list of products are displayed.

Verify the Service Request Creator Component

To verify the Service Request Creator component:

1. Sign in to your Digital Customer Service application as a self-registered user.
2. Click the **User** menu, then select **My Service Requests**.
3. Click **Create Service Request**.
4. Enter details in the following fields:
    - **Title**
    - **Describe the Problem**
    - **Category**
    - **Product**
5. Click **Submit**.
6. Verify that a confirmation message appears.

Verify the Service Request List Component

To verify the Service Request List component:

1. Sign in to your Digital Customer Service application as a self-registered user.
2. Click the **User** menu, then select **My Service Requests**.
3. Verify that the **Service Requests** list contains the service requests created in the Verify the Service Request Creator Component task.
4. Specify a **Filter**.
5. Verify that the **My Service Requests** list returns only SRs that contain the filter in the title.
6. Change the **Sort By** to **Service Request Number Ascending**.
7. Verify that the **Service Request** list is ordered by SR number, where the lowest SR number appears first in the list.

Verify the Edit Service Request Data Component

To verify the Edit Service Request Data component:

1. Sign in to your Digital Customer Service application as a self-registered user.

2. Click the **User** menu, then select **My Service Requests**.
3. View an existing SR by clicking on it from the **My Service Requests** list.
4. Verify that the details of the SR that you selected are displayed as expected.
5. Add a message:
    a. In the **Messages** tab, enter a message in the **Write a new message** field.
    b. Click **Submit**.
    c. Verify that the message you entered is displayed in the list of messages at the end of the page.
6. Add a file attachment:
    a. Click the **File Attachments** tab.
    b. Drag-and-drop files into the **Drop files to attach or browse** box.
    c. Click the **Pencil** (Edit Description) icon next to an uploaded file, enter a description, then click the **Check Mark** (Update Description) icon.
    d. Verify that the file and its description are listed in the list of file attachments.
    e. Click the file name.
    f. Verify that the file is successfully downloaded.
7. Add a URL attachment:
    a. Click the **URL Attachments** tab.
    b. Enter a URL and a description.
    c. Click **Submit**.
    d. Verify that the URL and its description are listed in the list of URL attachments.
8. Escalate the SR:
    a. Click the **Actions** menu and select **Escalate this request**.
    b. Enter a message in the **Escalate this request** dialog box.
    c. Click **Escalate**.
    d. Verify that the SR displays an escalated indicator.
    e. Verify that the escalation message appears in the list of messages on the **Messages** tab.
9. Close the SR:
    a. Click the **Actions** menu and select **Resolve this request**.
    b. Enter a message in the **Resolve this request** dialog box.
    c. Click **Resolve**
    d. Check that the SR displays a **Status** of **Resolved**.
    e. Verify that the closure message appears in the list of messages on the **Messages** tab.

Verify the Chat Component

To verify the Chat component:

1. Sign in to your Digital Customer Service application as a self-registered user.
2. Click the **Live Chat Support** link.
3. Fill in the **Subject** field.

    > **Note:** Some user details are prepopulated in the form.

4. Click **Start Chat**.
5. Verify that the chat connects, and is placed in the queue.

If the chat component isn't working, you may need to enable the SVC_ENABLE_CHAT profile option. For more information, see Related Topics for a link to the Configure Chat Profile Options topic.

Verify Search Results with Product and Category as Search Criteria

To verify search results where both product and category are used as search criteria:

1. Sign in to your Digital Customer Service application as a self-registered user.
2. Enter your search criteria in the **Search** text box.
3. Expand the **Filter your search** option.

   The **Select category** and **Select product** fields appear.
4. Navigate the product hierarchy in the **Select product** field, then click a product.
5. Navigate the category hierarchy in the **Select category** field, then click a category.
6. Click the **Magnifying Glass** (Search) icon.

   Your search results appear.
7. Verify the results match the search criteria.

Add a Fusion Service Field to the Service Request Page

There several steps you perform to add a custom field from Fusion Service into the Service Request form.

- Replace the service metadata: You first replace the service metadata in Digital Customer Service with the latest metadata from Fusion Service, including the configurations you would like the Digital Customer Service application to consume.
- Edit the page itself. You add the new field, change the REST API call to ensure that the new field is requested or sent, and then you add a new UI control for the new field onto the page.

Replace the Metadata

Replace the endpoint metadata in the application with the latest metadata from the server that includes the new configuration.

1. In Visual Builder select the **crmRestApi** in the Services window, and then click the **Service Connections** icon in the navigator.
2. Click the **Endpoints** tab.
3. Click the **Replace definitions for all selected endpoints** icon.
4. Click **Replace** on the **Confirm definitions replace** dialog box.
5. Once the operation is complete, click **OK**.

Add the New Field

Now you add the new field to the corresponding type in the Digital Customer Service application

1. In the navigator, click the **Web Application** icon.
2. Expand the **Flows** node.
3. Expand the **service-request-create** flow.

4. Select **service-request-create-start**.
5. From the Designer list, select the **Variables and Types** icon.
6. In the workspace, for the createServiceRequest type, click **Edit from Endpoint** .
7. From the **Endpoint Structure** list, select the custom field, and then click **Finish**.

Change the REST Request

Change the action chain for the REST request to ensure that the field is included in the body of the create request and mapped to the property on the object.

1. Ensure that the **service-request-create-start** is selected, and then from the Designer list, select the **Actions** icon.
2. In the workspace, select **SubmitServiceRequestAction**.
3. In the workspace, select the callCreateServiceRequestEndPoint REST call.
4. In the **Call Action Chain** window, in the **Input Parameters** area, click **Assign**.
5. In the **Call Action Chain** workspace, in the **Target** area, click **body**.
6. Before the closing brace in the JSON code, add a new entry for the custom field. Make sure you add a trailing comma to the preceding line.

   For example, `"MyCustomField_c": [[ $page.variables.serviceRequest.MyCustomField_c ]]`

Change the GET REST Request for the List

Edit the action chain or service data provider that does the GET request and make sure the field parameter of the GET includes the custom field. You also may need to map the field in the response to a variable.

1. In the Web Apps list, expand the Flows node, and then expand the **service-request-list** node.
2. Select **service-request-list-start**
3. In the Designer, click the **Variables and Types** icon.
4. In the Variables workspace, select **srListServiceDataProvider**.
5. In the **Parameters** list, select the **fields** parameter.
6. In the **urlParameters.field** area, add a comma at the end of the field list, and add the Fusion Service custom field name, then click **Save**.

Change the GET REST Request for the Detail Page

Now edit the request for the detail page.

1. From the Flows list, expand the **service-request-detail** node, and then select **service-request-detail-start**.
2. Click the **Actions** icon in the Designer.
3. Locate **LoadServiceRequestAction**, and click the **Menu** icon in that row, and select **Edit**.
4. In the diagram, click **Call REST Endpoint**.
5. In the **Input Parameters** list, select **fields**.
6. Add the new custom field to the comma separated list of fields and then click **Save**.

How do I enable my own sign in pages?

To enable your own sign-in pages, you must first update your Digital Customer Service application configuration to use Digital Customer Service as the Security provider.

After doing this, VBCS will inject data into the `app-flow.json` file which will allow the Digital Customer Service

Security Provider to have access to the same IDCS configuration information as the standard VB security provider. In addition to enabling you to create your own sign-in page, using the Digital Customer Service Security Provider enables you to use the Change Password button on the My Profile page in the out of the box reference implementation. That button takes the user to the reference implementation's Change Password page.

1. Using the application navigator's Source View, navigate to **webApps > webAppName > app-flow.json**.
2. Locate the `userConfig` element in the `app-flow.json` file of the application in the DT editor, and replace the child element `"type": "vb/DefaultSecurityProvider"` with : `"type: "oj-odcs/application-common/OdcsSecurityProvider".`
3. Then, add `"defaultSecurity": true.`

   This entry must be a child element of the `"userConfig/configuration"` element which tells VBCS to send IDCS configuration data to the `app-flow.json` file during application staging. Here's how the updated `"userConfig"` element should look:

```
"userConfig": {
 "type": "oj-odcs/application-common/OdcsSecurityProvider",
 "configuration": {
 "defaultSecurity": true,
 "authentication": {
 "type": "implicit"
 }
 },
 "embedding": "deny"
 }
```

You can optionally specify the custom sign-in page for Digital Customer Service by setting `userConfig.configuration.odcsLoginPath` to point to a VB page path. If this path isn't specified, the RI default of `shell/sign-in` is used.

Set the Service Instance URL

Now you must set the service instance URL for the idcsRestApi Service Connection.

1. In Oracle Visual Builder, open your Digital Customer Service application (if it's not open already).
2. Click Service Connections, click **idcsRestApi**, and then click the **Servers** tab.
3. Click **Edit**, then in the Edit Server page, edit the Instance URL to be the URL of your IDCS server. For example: `https://idcs-xxx.identity.yyy.idcs-example.com.`
4. Click **Save**.

Verify the Identity Cloud Service Identity Provider Policy

If you have a custom sign-in page for your Digital Customer Service application confirm that IDCS is enabled to use the User Name-Password authentication factor.

1. In the Identify Cloud Service administration console, click the Navigation Drawer, the expand **Security**, and choose **IDP Policies**.
2. Click **Default Identity Provider Policy**, and then the **Identity Provider Rules** tab.
3. Click the **Menu** icon and choose **Edit**.
4. In the Edit Default IDP Rule dialog box, make sure **Username-Password** is shown in the **Allowed Identity Providers** box. If not, do the following:

   a. Click in the **Assign Identity Providers** box, and select **Username-Password**.

b. Click **Save**.

If you don't want to add `Username-Password` to the Default Identity Provider Policy, you can add a new IDP Policy for the Digital Customer Service application to use. When you create the new policy, add a rule that allows the Username-Password Identity Provider and assign the VBINST_xxxxx application to the policy. This will ensure that the application uses the new IDP policy instead of the default IDP policy.

For more information, see Related Topics for a link to the Identity Cloud Service documentation.

Related Topics

- Understand Identity Provider Policies

How do I enable automatic Sign in for authenticated Identity Cloud Service users?

You can configure your custom sign-in page in the Digital Customer Service application to automatically log in users who are already authenticated through Identity Cloud Service.

Users can be authenticated either directly through the IDCS administration console, or through another application that uses the same IDCS instance as the Digital Customer Service application. This functionality is disabled by default, but can be enabled by setting the **enableAutoLogin** constant in the new **LoginAuthenticatedIdcsUser** application level action chain to **true**.

Here's how you do it:

1. In Visual Builder, click your application icon in the Web Apps navigator.
2. Click the **Actions** tab.
3. In the list of Actions, locate and select the **LoginAuthenticatedIdcsUser** action.
4. Set the default value of the **enableAutoLogin** constant to **true**.

How do I enable my own Forgotten Password page?

This topic is required to set up Identity Cloud Service to include the Digital Customer Service Reset Password page URL link when a user requests a password reset.

For this functionality to work Identity Cloud Service requires a REST request to be made to the `IDCS /Settings` API by a tenant administrator to set `allowedNotificationRedirectUrls` to include the Digital Customer Service Reset Password page URL. This setting defines the allowed notification redirect URLs which can be specified as the value of `notificationRedirectUrl` in the POST `. . . /admin/v1/MePasswordResetRequestor` request payload, which is then included in the reset password email notification sent to a user as part of the forgot password and password reset flow.

If you need to create the App in IDCS, see *Create a Client application*.

> **Note:** You must have Identity Administrator role and the Client ID and Client Secret must be for an IDCS application which has either the Identity Domain Administrator or Security Administrator application role granted. This is required to successfully call the `/Settings` REST API.

Use the following PowerShell script. If you're using a Mac or Linux computer, you'll need to install PowerShell first to run the script. See Microsoft's website for details. PowerShell is installed by default on Windows.

Save the following script in a file called `passwordreset.ps1` and change the `'...'` values in the script as appropriate for your environment.

- To run the script from a Command Prompt on Windows enter: `powershell -File passwordreset.ps1`
- To run the script on Mac or Linux enter: `pwsh passwordreset.ps1`

```
### MODIFY THE FOLLOWING VARIABLES FOR YOUR IDCS/ODCS ENVIRONMENT ###

# Set IDCS variables (modify for your IDCS instance)
$IdcsUrl = '...' # e.g. 'https://idcs-xxx.identity.yyy.idcs-example.com'
$ClientId = '...' # Client ID for privileged app in IDCS
$ClientSecret = '...' # Client Secret for privileged app in IDCS

# Either set the first 3 variables below for your ODCS app or explictly define $ForgotPasswordUrl
$OdcsHost = '...' # e.g. 'my-odcs-example.com'
$OdcsAppName = '...' # e.g. 'my_odcs_app'
$OdcsVersion = '...' # e.g. '1.1'
$ForgotPasswordUrl = "https://${OdcsHost}/ic/builder/rt/${OdcsAppName}/${OdcsVersion}/webApps/dcs/?
page=shell&shell=forgot-password"

### DO NOT MODIFY THIS SCRIPT BELOW THIS LINE ###

# Generate an access token
$Credentials = [Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes("${ClientId}:
${ClientSecret}"))
$Uri = "$IdcsUrl/oauth2/v1/token"
$Headers = @{
 Authorization = "Basic $Credentials"
}
$Parameters = @{
 grant_type = 'client_credentials'
 scope = 'urn:opc:idm:__myscopes__'
}
try {
 $Response = Invoke-RestMethod -Uri $Uri -Method POST -Headers $Headers -Body $Parameters
} catch {
 Write-Host ("Access token request POST to {0} failed with an error: {2}`nForm parameters: {1}`nException:
 {3}" -f $Uri, ($Parameters | Out-String), $_.ErrorDetails, $_.Exception) -fore red
 exit
}
$AccessToken = $Response.access_token

Write-Debug "Access Token = $AccessToken"

# Check the 'allowedNotificationRedirectUrls' IDCS Setting value
$Uri = "$IdcsUrl/admin/v1/Settings/Settings/?attributes=allowedNotificationRedirectUrls"
$Headers = @{
 Authorization = "Bearer $AccessToken"
 'Content-Type' = 'application/scim+json'
}
try {
 $Response = Invoke-RestMethod -Uri $Uri -Method GET -Headers $Headers
} catch {
 Write-Host ("Request to GET {0} failed with an error: {1}. `nException: {2}" -f $Uri, $_.ErrorDetails,
 $_.Exception) -fore red
 exit
}
$AllowedUrls = $Response.allowedNotificationRedirectUrls

Write-Debug "Allowed URLs from GET /admin/v1/Settings/Settings = $AllowedUrls"

# Add the forgot password URL to the Settings, if required (i.e. not already in $AllowedUrls)
```

```
if ($null -ne $AllowedUrls -And $AllowedUrls.Contains($ForgotPasswordUrl)) {
 Write-Output "URL ($ForgotPasswordUrl) is already registered"
} else {
 # Remove query parameter from /Settings URL, headers remain as for previous request
 $Uri = "$IdcsUrl/admin/v1/Settings/Settings/"

 # Add new URL and format the list for inclusion in the JSON payload (without powershell encoding)
 $AllowedUrls += $ForgotPasswordUrl;
 $AllowedUrls = '"{0}"' -f ($AllowedUrls -join '","')
 $Body = '{
 "schemas": [
 "urn:ietf:params:scim:api:messages:2.0:PatchOp"
 ],
 "Operations": [
 {
 "op": "replace",
 "path": "allowedNotificationRedirectUrls",
 "value": [' + $AllowedUrls + ']
 }
 ]
 }'

 try {
 $Response = Invoke-RestMethod -Uri $Uri -Method PATCH -Headers $Headers -Body $Body
 } catch {
 Write-Host ("Request to PUT {0} to {1} failed with an error: {2}. `nException:{3}" -f $Body, $Uri,
 $_.ErrorDetails, $_.Exception) -fore red
 exit
 }

 $AllowedUrls = $Response.allowedNotificationRedirectUrls
 Write-Output "Added new URL: $ForgotPasswordUrl"
}

Write-Output "`nAllowed Notification Redirect URLs:"
Write-Output $AllowedUrls
```

Localize Digital Customer Service for Multilingual Support

This topic describes how to localize Digital Customer Service for multilingual support.

How You Create Localized Digital Customer Service Applications

You can create localized versions of your application by translating the UI text and messages in your application into other languages. The localized strings are displayed in the application during runtime. When you run the staged or published application, a localized version is displayed based on the language settings of your browser.

A Digital Customer Service application created using the Reference Implementation template includes translations to the 25 languages supported by Oracle Fusion Service. While the Reference Implementation template is delivered with these translation strings and files, any modifications that you apply to the strings in your Digital Customer Service application will require extra translation.

The supported default language locales are as follows: **ar**, **cs**, **da**, **de**, **es**, **fi**, **fr-CA**, **fr**, **hu**, **it**, **iw**, **ja**, **ko**, **nl**, **no**, **nl**, **pt-BR**, **ro**, **ru**, **sk**, **sv**, **th**, **tr**, **zh-CN**, **zh-TW**.

The resource bundles provide an initial set of translated strings for most of the application that you can use for your own translation files. To populate the rest of the translation files for the languages you want to support, follow the instructions for generating files for new languages, and then use the existing translated files to obtain translations for use in your new file.

The Digital Customer Service Reference Implementation template comes with a single, application specific bundle. If you create strings, you must add the translations to the bundle, or create bundles.

For comprehensive information about adding translated text to your application, see the Work with Translations topic in the Develop Applications chapter of the Developing Applications with Oracle Visual Builder guide in the Related Topic.

Configure Languages Available in your Digital Customer Service Application

By default, the preferred language specified in the viewing browser is used to display your Digital Customer Service application to users. In the Digital Customer Service Reference Implementation template, a languages menu can be configured to list the specific languages that you want to make available to your users, should they want to use another language.

To configure the languages available to your Digital Customer Service application users:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click **Web Applications** .
4. In the **Web Apps** tree, expand `dcs`.
5. Click the **Variables** tab.
6. Select **localeList** from the **Constants** section.
7. Copy the language that you want to include in the languages menu from the **Description**.

   For example, to make Italian available, copy the following string: `{ code: "it", name: "Italiano"}`
8. Add it to the default value ensuring that the JSON is correctly formatted, commas between entries in the array and no trailing comma after the last entry.
9. Click the **Run** icon to view the change in your application.
10. Click the **Globe** icon (Languages) to verify that the language that you added appears in the list of available languages.

Set a Non English Default Language

By default the starting language for your application is English, to change this there you must change variable values. Use the list of supported locales outlined earlier in this topic for the correct format. For this exercise, we'll use Spanish as an example:

1. In the `index.html` page, do the following: find the following entry:
   a. Find the following entry: `<html lang="en">`
   b. Change the entry to: `<html lang="es">`.
   c. Now, find the var lang entry: `var lang =window.localStorage.getItem('odcs-reference-implementation.locale') || navigator.language;`
   d. Change the entry to: `var lang = window.localStorage.getItem('odcs-reference-implementation.locale') || "es";`
2. Now, in the `app-flow.json` file, and do the following:
   a. Find the entry to:`"{{ window.localStorage.getItem('odcs-reference-implementation.locale') || navigator.language }}"`

b. Change that entry to: `"locale": "{{ window.localStorage.getItem('odcs-reference-implementation.locale') || 'es' }}"`

3. The default language is now Spanish. Use these examples and substitute your locale code as needed.

Create Translations for New Languages

You can introduce a new language beyond the ones delivered with a Digital Customer Service application that was created using the Reference Implementation.

For more information, see the Work with Translations topic in the Developing Applications with Oracle Visual Builder in the Related Topics.

To introduce a new language beyond the ones delivered with a Digital Customer Service application that was created using the Reference Implementation, follow these steps:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Menu** icon in the editor, and select **Settings**.

   The **Settings** tab appears.
4. Click the **Translations** secondary tab.
5. Download the latest language bundle by clicking the `ARB` or `XLIFF` download link in the **Download all strings** section.

   A `zip` file is downloaded.
6. Open your **zip** file.
7. Extract the **app-strings** file.
8. Rename the **app-strings** file to **app-strings-<locale>.arb** or **app-strings-<locale>.xliff**, depending on the downloaded file type. Where `<locale>` is the language locale.

   For example, for a file containing Welsh translations, the change file name should be: `app-strings-cy.arb` or `app-strings-cy.xliff`.
9. Open your **app-strings-<locale>** file in a text editor. The first few lines of the file will resemble the following:

```
{
 "@@x-bundleName" : "app",
 "@@x-bundlePath" : "webApps/dcs/resources/strings/app/nls/app-strings",
 "common_save" : "Save",
 "@common_save" : {
 "description" : "Used by components that perform a saving operation.",
 "source_text" : "Save"
 },
 "common_cancel" : "Cancel",
 "@common_cancel" : {
 "description" : "Used by components that cancel the current operation.",
 "source_text" : "Cancel"
 },
 "common_done" : "Done",
 "@common_done" : {
 "description" : "Used by components that finish an operation causing no changes.",
 "source_text" : "Done"
 },
 "common_download" : "Download",
 "@common_download" : {
 "description" : "Used by components that require a download label.",
 "source_text" : "Download"
```

10. Replace the English strings with the translation for the language you're creating. Consider the following `Cancel` string:

```
},
"common_cancel" : "Cancel",
"@common_cancel" : {
"description" : "Used by components that cancel the current operation.",
"source_text" : "Cancel"
},
```

    You'd want to replace the string after the `"common_cancel": "` with the translated language equal. In this example, you'd want to replace the string `Cancel` with the Welsh equal for Cancel, which is `Canslo`. The changed text will look similar to the following:

```
},
"common_cancel" : "Canslo",
"@common_cancel" : {
"description" : "Used by components that cancel the current operation.",
"source_text" : "Cancel",
},
```

11. Repeat step 10 for all the strings in the language that you're creating.
12. Save your **app-strings-<locale>** file.
13. Add the **app-strings-<locale>** file to the `zip` file that you downloaded in step 5, and save the updated `zip` file.
14. Click the **Menu** icon in the editor, and select **Settings**.
    The **Settings** tab appears.
15. Click the **Translations** secondary tab.
16. Upload the `zip` file that you saved in step 13, containing your new **app-strings-<locale>** file to the **Uploaded updated bundle** section.
    The language is added to the bundle.

Add New String Translations for Existing Languages

You can translate more strings after changing strings in your Digital Customer Service application generated from the Reference Implementation template.

For more information, see the Work with Translations topic in the Developing Applications with Oracle Visual Builder in the Related Topics.

To add new string translations for existing languages:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Menu** icon in the editor, and select **Settings**.
    The **Settings** tab appears.
4. Click the **Translations** secondary tab.
5. Download the latest language bundle by clicking the `ARB` or `XLIFF` download link in the **Download new and changed strings** section.
    A `zip` file is downloaded.
6. Open your **zip** file.
7. Extract the **app-strings** file.
8. Rename the **app-strings** file to **app-strings-<locale>.arb** or **app-strings-<locale>.xliff**, depending on the downloaded file type. Where `<locale>` is the language locale.

x

For example, for a file containing French translations, the changed file name should be: `app-strings-fr.arb` or `app-strings-fr.xliff`.

9. Open your **app-strings-<locale>** file in a text editor.
10. Replace the English strings with the translation for the language you're updating. Consider the following `Submit Now` string:

```
},
"common_submit_now" : "Submit Now",
"@common_submit_now" : {
"description" : "Used to submit data entered in a form immediately.",
"source_text" : "Submit Now",
},
```

You'd want to replace the string after the `"common_submit_now": "` with the translated language equal. In this example, you'd want to replace the string `Submit Now` with the French equal for Submit Now, which is `Soumettre Maintenant`. The changed text will look similar to the following:

```
},
"common_submit_now" : "Soumettre Maintenant",
"@common_submit_now" : {
"description" : "Used to submit data entered in a form immediately.",
"source_text" : "Submit Now",
},
```

11. Repeat step 10 for all the strings in the language that you're creating.
12. Save your **app-strings-<locale>** file.
13. Add the **app-strings-<locale>** file to the `zip` file that you downloaded in step 5, and save the updated `zip` file.
14. Click the **Menu** icon in the editor, and select **Settings**.

The **Settings** tab appears.

15. Click the **Translations** secondary tab.
16. Upload the `zip` file that you saved in step 13, containing your new **app-strings-<locale>** file to the **Uploaded updated bundle** section.

The language bundle is updated with the new string translations for the existing language.

Related Topics

- Developing Applications with Oracle Visual Builder
- Work with Translations

Propagate Attribute Changes of Fusion Service Contacts to IDCS Users Accounts

Overview of Steps to Propagate Attribute Changes of Fusion Service Contacts to IDCS Users Accounts

When you use Self-Service Optimization, a self-service user is represented by a Contact record in the Fusion Service application and by a User account in Identity Cloud Service.

This topic shows you how changes made to attributes of a contact record can be propagated to the user account associated with this contact in IDCS.

When the attributes of a contact are changed, an object workflow is triggered. You can use this workflow to propagate any changes to attributes of interest to a user account in IDCS. Here's a high level overview of the required steps:

- Check if the contact record is that of a self-service user.
- Verify if the attributes of interest changed.
- Use the Party ID of the contact to find a Self-Service Role record.
- Obtain the Login ID of the user from the Self-Service Role record.
- Use the Login ID to find the GUID of the user account in IDCS.
- Change the attributes of interest in IDCS

Setup for Coding

Here are the setup tasks you need to perform to prepare for the required coding.

1. Sign in to Fusion Service.
2. Click Navigator, and expand Tools.
3. Select **Security Console**.
4. Click the **API Authentication** tab.
5. Click the **Oracle Public Certificate** link to download the **orakey_sign** certificate.

Create a Trusted Client Application in Identity Cloud Service

Use this task to create a trusted client application in IDCS.

1. In the IDCS admin console, click the **Navigator** menu and choose **Applications**.
2. Click **Add**.
3. In the dialog box that appears, choose **Confidential Application**.
4. In the **Name** field enter **Contact Sync App** and click **Next**.

5. Select **Configure this application as a client now**.
6. Choose the following Allowed Grant Types:
    - Client Credentials
    - JWT Assertion
7. Click the **Trusted Client** check box.
8. Click **Import**.
9. Enter **OraKeySign** in the **Certificate Alias** field.
10. Choose the certificate file that was downloaded in the previous step and click **Import**.
11. Click **Add** in the **Grant the client access to Identity Cloud Service Admin APIs** workspace.
12. In the **Add App Role** dialog box, choose **User Administrator** and click **Add**.
13. Click **Next** to advance to the **Resources** tab stop.
14. Click **Next** to advance to the **Web Tier Policy** tab stop.
15. Click **Next** to advance to the **Authorization** tab stop.
16. Click **Finish**.
17. Record the **Client ID** and **Client Secret** and then dismiss the dialog box.
18. Click **Activate** and then **OK** to activate the client application.

Create a User

Now you use Identity Cloud Service to create a user.

1. Sign in to IDCS and click the Navigation drawer, and select **Users**.
2. Click the Add (**+**) icon.
3. In the **Add User** window, enter the following:
    - First Name: contact.sync
    - Last Name; admin.user
    - User Name: contact.sync.admin.user

      > **Note:** Deselect the **Use the email address as the user name** check box.

    - Email: Enter the user's email address.
4. Click **Finish**.

   A Welcome email will now be sent to the email address you entered in this procedure. This email provides instructions on how to change the password.

Assign the User Administrator Role

Now, you assign the User Administrator role to the newly created user.

1. Sign in to IDCS and click the Navigation drawer, and select **Administrators**.
2. Expand **User Administrator**.
3. Click the Add (**+**) icon.
4. In the **Add Users to the Administrator Role** work area, enter **contact.sync.admin.user** in the search field.
5. Select the sync.admin user, and click **OK** to add the User Administrator role.

Create Web Services

Here's the initial setup for these tasks:

1. Sign in to Fusion Service and activate a sandbox.
2. Click **Navigator > Configuration > Application Composer**.
3. In the Explorer, expand **Common Setup**, then click **Web Services**.

Create the FindIdcsGuidByLoginId Web Service

Create the FindIdcsGuidByLoginId web service by doing the following:

1. In the Web Services work area, click the **Create Web Service reference** icon.
2. In the Select Connection Type dialog box, choose **REST** and then click **OK**.
3. In the **Create REST Web Service Connection** work area, enter the following in the **Name** field: **FindIdcsGuidByLoginId**.
4. In the **URL** field enter

   ```
   https://<IDCS_HOST>>/admin/v1/Users?filter=userName%20eq%20%22##LOGIN_ID##%22
   ```

5. In the Authentication Scheme area, select **Call using IDCS OAUTH**.
6. Click the Create Credential Key icon (**+**) beside the Client Credential Key drop down list.
7. In the **Client Key** dialog box, enter the following:
   - CSF Key: contact-sync-app-key
   - User Name: Enter the Client ID.
   - Password: Enter the Client Secret.
8. Click OK.
9. In the **Token URL** field enter the following:

   ```
   https://<<IDCS_HOST>>/oauth2/v1/token
   ```

10. In the **Scope** field enter `urn:opc:idm:__myscopes__`
11. Make sure the Subject Precedence check box isn't selected.
12. In the Authentication Scheme area, select **Call using IDCS OAUTH**.
13. Click the Create Credential Key icon (**+**) beside the Credential Key for Switch Identity drop down list.
14. In the CSF Key text box enter contact-sync-admin-key
15. In the **Client Key** dialog box, enter the following: field enter:
    - CSF Key: contact-sync-admin-key
    - User Name: Enter contact.sync.admin.user.
    - Password: Enter the password for the user.
16. Click **OK**.
17. In the **Select and configure Methods against the Resource** work area, select the **GET** check box.
18. For **Request Payload**, select the **Schema URL** option.
19. For **Response Payload** choose **Code Sample**, and enter the following in the code field: `{}`.
20. Click **Save and Close**.

Create the UpdateIdcsUserAttributes Web Service

Create the UpdateIdcsUserAttributes web service by doing the following:

1. In the Web Services work area, click the **Create Web Service reference** icon.
2. In the Select Connection Type dialog box, choose **REST** and then click **OK**.
3. In the **Create REST Web Service Connection** work area, enter the following in the **Name** field: **UpdateIdcsUserAttributes**.
4. In the **URL** field enter

   ```
   https://<IDCS_HOST>>/admin/v1/Users/##USER_ID##
   ```

5. In the Authentication Scheme area, select **Call using IDCS OAUTH**.
6. Click the **Client Credential Key** drop down list, and select: **contact-sync-app-key**.
7. In the **Token URL** field enter the following:

   ```
   https://<<IDCS_HOST>>/oauth2/v1/token
   ```

8. In the **Scope** field enter `urn:opc:idm:__myscopes__`
9. Make sure the Subject Precedence check box isn't selected.
10. Click the **Credential Key for Switch Identity** drop down list, and select: **contact-sync-admin-key**.
11. In the **Select and configure Methods against the Resource** work area, select the **PATCH** check box.
12. For **Method Name**, click the drop down list and select **PATCH**.
13. For **Format**, click the drop down list and select **JSON**.
14. For **Request Payload**, select **Code Sample**, and enter the following in the code field: `{}`
15. For **Response Payload** choose **Code Sample**, and enter the following in the code field: `{}`.
16. Click **Save and Close**.

Create the FindSelfServiceUser Web Service

Create the FindSelfServiceUser web service by doing the following:

1. In the Web Services work area, click the **Create Web Service reference** icon.
2. In the Select Connection Type dialog box, choose **REST** and then click **OK**.
3. In the **Create REST Web Service Connection** work area, enter the following in the **Name** field: **FindSelfServiceUser**.
4. In the **URL** field enter

   ```
   https://<<FA_HOST>>/crmRestApi/resources/11.13.18.05/selfServiceRoles?
   q=ContactPartyId=##CONTACT_PARTY_ID##%20and%20RelationshipTypeCd=%27ORA_CSS_USER%27
   ```

5. In the Authentication Scheme area, select **Call with Basic Authentication**.
6. Create or use a Client Credential Key for the user in Fusion Service that has the Sales Administrator role or Customer Self-Service Administration duty role.
7. In the **Select and configure Methods against the Resource** work area, select the **GET** check box.
8. For **Method Name**, click the drop down list and select **GET**.
9. For **Format**, click the drop down list and select **JSON**.
10. For **Request Payload**, select **Schema URL**.
11. For **Response Payload** choose **Code Sample**, and enter the following in the code field: `{}`.
12. Click **Save and Close**.

Create Global Functions

Now you create the following global functions that will be used in an object workflow groovy script.

Create the findSelfServiceUserLoginId Global Function

Use this topic to create the findSelfServiceUserLoginId global function.

1. In Application Composer, expand **Common Setup**, and select **Global Functions**.
2. Click the **Add a Global Function** icon.
3. Use the following table to fill in the necessary fields:

Field	What You Do
Function Name	Enter: `findSelfServiceUserLoginId`
Returns	Click the drop-down list and select **String**.
Parameters	a. Click to expand the workspace. b. Click the **Add Parameter** icon. c. In the **Name** field enter: contactPartyId. d. From the **Type** drop-down list, select **String**.

4. In the Edit Script field, copy and paste the following script:

```
def request = adf.webServices.FindSelfServiceUser

try{
 def httpHeaders=['REST-Framework-Version':'3']
 request.requestHTTPHeaders = httpHeaders

 def searchResults = request.GET(contactPartyId)
 def count = searchResults.get("count")

 if(count == 0){
 def message = "Unable to locate self-service user by Party ID: " + contactPartyId + ":" + searchResults
 throw new oracle.jbo.ValidationException(message);
 }

 def loginId = searchResults.get("items")[0].get("LoginId")

 return loginId
}catch(Exception e){
 throw new oracle.jbo.ValidationException(" " + e + ": " + request.httpErrorResponse)
}
```

5. Click **Save and Close**.
6. Click **Yes** to accept the warning message.

Create the getIdcsUserGuid Global Function

Use this topic to create the getIdcsUserGuid global function.

1. In Application Composer, expand **Common Setup**, and select **Global Functions**.
2. Click the **Add a Global Function** icon.
3. Use the following table to fill in the necessary fields:

Field	What You Do
Function Name	Enter: `getIdcsUserGuid`
Returns	Click the drop-down list and select **String**.
Parameters	a. Click to expand the workspace. b. Click the **Add Parameter** icon. c. In the **Name** field enter: loginId. d. From the **Type** drop-down list, select **String**.

4. In the Edit Script field, copy and paste the following script:

```
def request = adf.webServices.FindIdcsGuidByLoginId

try{
 def searchResults = request.GET(loginId)
 def totalResults = searchResults.get("totalResults")

 if(totalResults != 1){
 throw new oracle.jbo.ValidationException("Unable to locate user by login ID: " + loginId + ", total
 results: " + totalResults);
 }

 def resources = searchResults.get("Resources")
 def user = resources[0]
 def guid = user.get("id")

 return guid;
}catch(Exception e){
 throw new oracle.jbo.ValidationException("Error finding IDCS User: " + e + ":"
 +request.httpErrorResponse);
}
```

5. Click **Save and Close**.
6. Click **Yes** to accept the warning message.

Create the updateUserAttributesInIdcs Global Function

Use this topic to create the getIdcsUserGuid global function.

1. In Application Composer, expand **Common Setup**, and select **Global Functions**.
2. Click the **Add a Global Function** icon.

3. Use the following table to fill in the necessary fields:

Field	What You Do
Function Name	Enter: **updateUserAttributesInIdcs**
Returns	Click the drop-down list and select **void**.
Parameters	a. Click to expand the workspace. b. Click the **Add Parameter** icon. c. In the **Name** field enter:userId. d. From the **Type** drop-down list, select **String**. e. Click the **Add Parameter** icon. f. In the **Name** field enter: firstName. g. From the **Type** drop-down list, select **String**. h. Click the **Add Parameter** icon. i. In the **Name** field enter: lastName. j. From the **Type** drop-down list, select **String**.

4. In the Edit Script field, copy and paste the following script:

```
def conn = adf.webServices.UpdateIdcsUserAttributes
try{
def patch_body = [
 "schemas": [
 "urn:ietf:params:scim:api:messages:2.0:PatchOp"
 ],
 "Operations": [
 [
 "op": "replace",
 "path": "name",
 "value":
 [
 "givenName": firstName,
 "familyName": lastName
 ]
 ]
 ]
]
conn.PATCH(userId, patch_body)
}catch(Exception e){
 throw new oracle.jbo.ValidationException(" " + e + " " + conn. httpErrorResponse)
}
```

5. Click **Save and Close**.
6. Click **Yes** to accept the warning message.

Create the syncContactAttributes Global Function

Use this topic to create the getIdcsUserGuid global function.

1. In Application Composer, expand **Common Setup**, and select **Global Functions**.
2. Click the **Add a Global Function** icon.

3. Use the following table to fill in the necessary fields:

Field	What You Do
Function Name	Enter: `syncContactAttributes`
Returns	Click the drop-down list and select **void**.
Parameters	a. Click to expand the workspace. b. Click the **Add Parameter** icon. c. In the **Name** field contactPartyId. d. From the **Type** drop-down list, select **String**. e. Click the **Add Parameter** icon. f. In the **Name** field enter: firstName. g. From the **Type** drop-down list, select **String**. h. Click the **Add Parameter** icon. i. In the **Name** field enter: lastName. j. From the **Type** drop-down list, select **String**.

4. In the Edit Script field, copy and paste the following script:

```
def loginId = adf.util.findSelfServiceUserLoginId(contactPartyId)
def userGuid = adf.util.getIdcsUserGuid(loginId)
adf.util.updateUserAttributesInIdcs(userGuid,firstName,lastName)
```

5. Click **Save and Close**.
6. Click **Yes** to accept the warning message.

Create an Object Function on the Contact Object

Use this topic to create an object function on the contact object.

1. In Application Composer, expand the **Standard Objects** node, and then expand **Contact**.
2. Click Server Scripts, and then click the **Object Functions** tab.
3. Click the **Add a new Object Function** icon.
4. Use the following table to create the function:

Field	What You Do
Function Name	Enter: `syncContactAttributes`
Visibility	Click the drop down list and select: **Callable By External Systems**.

5. In the Edit Script field, copy and paste the following script:

```
def contactPartyId = PartyId + ""
def firstName = PersonFirstName
def lastName = PersonLastName
```

```
adf.util.syncContactAttributes(contactPartyId, firstName, lastName)
```

6. Click **Save and Close**.

Create Object Workflow on the Contact Object

Use this topic to create an object workflow on Contact object.

1. In Application Composer, expand **Common Setup**, then select **Object Workflows**.
2. Click the Create (**+**) icon.
3. Click the **Object** drop down list, and select **Contact**.
4. In the **Name** field, enter: **SyncContactAttributesToIdcsWf**.
5. For **Event Point and Condition**, select: **When a record is updated**.
6. Click in the Condition field, then click the Groovy builder icon.
7. Copy and paste the following code in the Expression Builder, then click OK.

```
if (ContactRole == 'SERVICE' &&
 (isAttributeChanged('PersonFirstName') || isAttributeChanged('PersonLastName')) )
 return true;
else
 return false;
```

8. Click the **Create** button of the Groovy Script Action.
9. In the **Name** field, enter: `SyncContactAttributesToIdcsGs`.
10. In the **Method Name** drop down list, select `SyncContactAttributes`.
11. Click **Save**, and then click **Save and Close**.

Test the Global Methods using Triggers

Now create triggers to test your global methods.

1. In Application Composer, expand **Standard Objects > Contact > Server Scripts.**.
2. Click the **Triggers** tab.
3. Click the **Add a new Trigger** icon.
4. Click the **Trigger** drop down list and select **Before Update in Database from the Trigger**.
5. In the **Trigger Name** field, enter: `test_sync_contact_attributes`.
6. Copy the following code and paste it in the Trigger Definition field:

```
def contactPartyId = PartyId + ""
def firstName = PersonFirstName
def lastName = PersonLastName
adf.util.syncContactAttributes(contactPartyId, firstName, lastName)
```

7. In the Contacts UI of the Fusion Service application, make a change to any self-service contact and make sure the script worked.
8. Once your testing and corrections are made, you can remove this trigger.

Test the Object WorkFlow

Update the First Name or Last Name of any self-service contact. Use Fusion Service administrator's console to verify if the changes to the contact record in Fusion Service were propagated to IDCS.

7 Users and Accounts

Self-Service Registration

Here's an overview of user self-service registration.

When the Digital Customer Service Reference Implementation template is used to create your Digital Customer Service application, users can take advantage of the self-service registration feature available to anonymous users through the **Sign-Up** link on the home page of their Digital Customer Service application. There are various profile options that control the way in which the self-service registration feature behaves. For more information about profile options relating to self-service registration, refer to Configure Profile Options in the related topics.

One profile option to note though is the ZCA_CONTACT_ADDRESS_REQUIRED_ENABLED profile option. You must set this profile option to one of two values:

- **No**. This means a contact address for the contact is optional.
- **Yes for Customer only**. This is the default value and it means that an address is required for contacts of the Customer type only.

> **Note:** The Digital Customer Service Self Service Registration feature doesn't support requiring addresses for all contact types, just for the Customer type.

The APPID user is used to call the self-registration API on behalf of the anonymous user when they use the **Sign-Up** link. For this API call to function the user must enter a unique email address.

> **Note:** A user can register multiple times if they need to be registered as a user in multiple customer accounts. Each registration request should use a different Account Key.

Related Topics

- How do I configure DCS profile options?

How do I set up Self-Service Registration without an account key requirement?

Administrators can enable this feature in the Setup and Maintenance work area. This feature is delivered enabled.

> **Note:** After enabling the self-service registration without an account key requirement, as specified in this section, the ODCS application developer specifies whether or not to require the account key by specifying the ODCS application-level variable userRegistrationType. The value **consumer** is used if you don't want to specify an account key. The default is **contact**, which requires an account key.

1. Sign in to Fusion Service as an administrator or setup user.
2. In the Setup and Maintenance work area, go to the following:
    - Offering: Service.
    - Functional Area: Digital Customer Service.
    - Task: Manage Digital Customer Service Profile Options.
3. Locate the SVC_CSS_ALLOW_CONSUMER profile option and set it to **Yes**.
4. **Optional step**: Locate the SVC_CSS_CONSUMER_USER_CATEGORY and set it to the user category for consumers which defines the redirect URL for self-service users after a password reset.
   After these settings are enabled, the Account key field will no longer be required for self-service users to register in the ODCS application.

Assign Roles to Digital Customer Service Users

This topic shows you how to assign user roles in the Digital Customer Service application.

Self-Service Account Administrator

Each customer account must have at least one self-service account administrator. You can assign or remove the account administrator role to user accounts in Oracle Fusion Service.

> **Note:** The first user created under a customer account is always automatically made granted the Account Administrator role for that account. The steps that are listed in this topic, are for creating additional account administrators.

To assign the self-service account administrator role:

1. Sign into Oracle Fusion Service as an administrator or setup user.
2. Navigate to the **Service** work area and click **Self-Service Users**.
3. From the **Self-Service Users** list, select the user you want to modify.
4. In the **User Administration** section, select the following role: **Account Administrator**.
5. Click **Save**.
   A dialog appears, confirming the role modification.

Self-Service Account Manager

You can assign the self-service account manager role to user accounts in Oracle Fusion Service.

To assign the self-service account manager role:

1. Sign into Oracle Fusion Service as an administrator or setup user.
2. Navigate to the **Service** work area and click **Self-Service Users**.
3. From the **Self-Service Users** list, select the user you want to modify.
4. In the **User Administration** section, select the following role: **Account Manager**.
5. Click **Save**.
   A dialog appears, confirming the role modification.

Create and Manage Custom Self-Service Roles

Custom self-service roles give you the flexibility to segment self-service users into different groups and provide them with a more specifically tailored experience.

For example, you might want to allow some users to see knowledge that other users can't see.

Digital Customer Service includes three self-services roles: User, Account Administrator and Account Manager. All self-service users are given the User role. The first user of an account is assigned the Account Administrator role. The following table shows the job roles these self-service roles are given and if they're used in data security policies:

Self-Service Role	Related Identity Provider Role	Use in Data Security Policy
User	Customer Self-Service User	No
Account Administrator	Customer Self-Service Account Administrator	Yes
Account Manager	No related role	Yes

Custom self-service roles can be defined and mapped to custom identity provider roles or used in data security policies. When a self-service user is assigned a custom self-service role, this user is also assigned to the corresponding custom identity provider role.

A custom self-service role can be mapped to only one custom identity provider role. Ready to use self-services roles can't be mapped to any identity provider roles.

Here's a list of tasks that you must perform to grant a self-service user a custom self-service role:

- Create a custom self-service role
- Create or find a role in Fusion Service
- Create or find a role in Identity Cloud Service
- Map the custom self-service role to an identity provider role
- Grant the custom self-service role to a self-service user

To assign these custom roles to new users, see "Add Mappings to User Roles" in Related Topics.

Create a Custom Self-Service Role

You can define custom self-service roles by adding a new code to the ORA_SVC_CSS_REL_TYPE_CD lookup type. You do this using the **Manage Self-Service Relationship Type Standard Lookup** task in Functional Setup Manager.

1. Sign in to Fusion Service as an administrator or setup user.
2. In the Setup and Maintenance area, go to the following:
    - Offering: Service.
    - Functional Area: Digital Customer Service.
    - Task: Manage Self-Service Relationship Type Standard Lookup

> **Tip:** Select **All Tasks** from the **Show** drop-down list to display the task.

3. Click **Manage Self-Service Relationship Type Standard Lookup**.
4. Add the custom self-service roles by adding the new codes to this standard lookup by doing the following:
    a. Click the New (**+**) icon.
    b. Enter values for the **Lookup Code** and **Meaning** fields.
    Optionally, enter values for other fields.
    c. Add a new row for each new self-service role.
    d. Click **Save and Close**.

Create or Find a Custom Identity Provider Role in Fusion Service Using Security Console

For this task you create a new role or find an existing role in Security Console. For more information on creating roles, see the Create Job and Abstract Roles topic from the Securing CX Sales and Fusion Service guide, in the Related Links

Create or Find a Custom Identity Provider Role in Identity Cloud Service

The custom role must exist in Identity Cloud Service also and the code of the role in Fusion Service must be the external ID of the group in Identity Cloud Service. You use IDCS SCIM API calls can to create the group in IDCS. See the following example of the API Call.

```
curl --location --request POST 'https://idcs-002d5462cc68420a96fcb9ed392854d2.identity.c9dev2.oc9qadev.com/
admin/v1/Groups' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJ4NXQjUzI1NiI6InZfUW8.....LCZhakQ' \
--data-raw '{
"displayName": "Adjudicator",
"externalId": "CUST_ADJUDICATOR",
"urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group": {
"creationMechanism": "api",
"description": "Created using Postman"
},
"schemas": [
"urn:ietf:params:scim:schemas:core:2.0:Group",
"urn:ietf:params:scim:schemas:oracle:idcs:extension:group:Group",
"urn:ietf:params:scim:schemas:extension:custom:2.0:Group"
]
}'
```

For more information on creating groups, see the Create Groups in Oracle Identity Cloud Service topic in the Related Topics.

Map the Custom Self-Service Role to an Identity Provider Role

For this task you use the Functional Setup Manager task, Manage Custom Role Mapping for Digital Customer Service to enable the viewing of existing mappings and creating new mappings between custom Self-Service roles and custom identity provider roles.

1. Sign in to Fusion Service as an administrator user.
2. In the Setup and Maintenance screen, select **Service**, then **Digital Customer Service**.
3. From the Show drop-down list, select **All Tasks**.
4. Click **Manage Custom Role Mapping for Digital Customer Service**.
5. Select a Self-Service role.

6. Select a Job role.
7. Click **Save**.

Grant a User the Custom Self Service Role

You use the selfServiceRoles REST API to assign a custom self-service role to a self-service user. The functional security required to use this API is given to the Customer Self-Service Account Administrator job role and the Customer Self-Service Administration duty role.

Here's an example:

```
curl --location --request POST 'https://<POD>.fa.<data center ID>.oraclecloud.com/crmRestApi/
resources/11.13.18.05/selfServiceRoles' \
--user 'ADMIN_USER:ADMIN_USER_PWD' \
--header 'Content-Type: application/json' \
--data-raw ' {
"AccountPartyId": 100000015022002,
"ContactPartyId": 300100544667497,
"RelationshipTypeCd": "CUST_ADJUDICATOR"
}'
```

Related Topics

- Create groups in Fusion Service

Grant Self-Service Access to Resources

You can grant self-service access to employees who are resources in Fusion service using the selfRegistrations REST resource provided those employees have been given the CONTACT usage.

To grant Contact usage to these resources, you use the Trading Community Party Information Service.

The following table shows the relevant information for this SOAP web service:

Requirement	Enter this value
Service WSDL URL	https://<Fusion Host>/crmService/PartyInformationService?WSDL
Operation	assignPartyUsage
Required Parameters	• PartyId • PartyUsageCode = CONTACT • CreatedByModule = FUSE

A UI widget you create such as a button can be configured to execute this groovy code. For more information refer to the Actions and Links topic of the Configuring Applications Using Application Composer documentation shown in Related Topics.

Here's an example of a groovy script that invokes the assignPartyUsage operation of the Party Information Service SOAP API.

```
def curPartyId = <Party ID of Resource>
Date curDate = new Date()

def partyUsageAssignment =
[
 PartyId : curPartyId,
 PartyUsageCode : 'CONTACT',
 EffectiveStartDate : curDate,
 CreatedByModule : 'FUSE'
 ]

adf.webServices.Call_Party_Usage_Service.assignPartyUsage(partyUsageAssignment)
```

Set Up Customer Accounts

Create a custom Account Key field

Use this topic to create a custom Account Key field for Digital Customer Service.

The value of the new Account Key field must be unique to each account and can be set to any value. If the value is set to a string that's easy to guess then someone could guess the value and create for themselves a user account. This is a particular problem if automatic approval is set to true. If the account key is set to a string that's difficult to guess then only those to whom the string is sent can successfully submit a registration request.

End user customers are required to have a known account key. Users must also specify an appropriate account key when they register for a self-service user account.

Multiple tasks must be completed to create a custom Account Key field for Digital Customer Service. To create a custom Account Key field for Digital Customer Service, complete these tasks in the order that they appear in this topic:

1. Add the Account Key to the Account Object
2. Add the Account Key to Pages
3. Populate the Account Key for Existing Accounts
4. Publish the Sandbox
5. Set a Profile Attribute for the Account Key

Add the Account Key to the Account Object

To add a custom field for the Account Key to the Account object, you'll need to use the Application Composer in Oracle Fusion Service.

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Create a sandbox for adding the Account Key field:
    a. Click **Navigator > Configuration > Sandboxes**.
    b. Click the **+** icon (**New**) to create the new sandbox.

       The **Create Sandbox** dialog appear.
    c. Enter a name in the **Sandbox Name** field.
    d. Click **Save and Close**.

       A confirmation dialog appears.
    e. In the **Manage Sandboxes** list, click the line item with the sandbox name that you specified in step c.

f. Click **Set as Active**.

3. Navigate to the **Application Composer**.
4. Expand **Objects**, then **Standard Objects**, then **Account**, and then click **Fields**.

   The **Select Field Type** page appears.
5. Click the **Action** menu, and select **Create**.

   The **Select Field Type** dialog appears.
6. Click the **Text** option, then click **OK**.

   The **Create Text Field** page appears.
7. Specify the following for the Account Key text field options:
   - In the **Display Label** field, enter the following string:

     `Account Key`
   - The **Name** field will be pre populated based on the name that you entered for the **Display Label**, without any spaces.
   - The **API Name** field will be pre populated based on the name that you entered for the **Display Label**, without any spaces, and typically with the following suffix: `_c`

     > **Tip:** You'll need to note the value assigned to the **API Name** field, because it will be assigned to the `SVC_CSS_ACCT_KEY_FIELD` profile option in the Set a Profile Attribute for the Account Key task, later in this topic.

   - In the **Display Type** option, click the **Simple Text Box** option.
   - Deselect the **Required** option in the **Constraints** section.
   - Select the **Updatable** option in the **Constraints** section.
   - Select the **Searchable** option in the **Constraints** section.
   - Select the **Include in Service Payload** option in the **Constraints** section.
8. Click **Save and Close**.

Add the Account Key to Pages

In this task, you'll add the Account Key field that you created in the previous task to the necessary pages in the Oracle Fusion Service Application Composer.

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Navigate to the **Application Composer**.
3. Expand **Objects**, then **Standard Objects**, then **Account**, and then click **Pages**.

   The **Account: Pages** page appears.
4. In the **Creation Page Layouts** section, select a layout to base your page on, such as **Standard layout**.
5. In the **Creation Page Layouts** section, click the **Actions** menu, and select **Duplicate**.

   The **Duplicate Layout** dialog appears.
6. Enter a name in the **New Layout Name**.
7. Click **Save and Edit**.
8. Click the **Pencil** icon (**Edit**) next to `FUSE Customer ObjectCreation View`.
9. The **Creation Layout** page appears.

10. From the **Available Fields** column, click the `AccountKey` field that you created in the previous task, then click the arrow to move it to the **Selected Fields** column.
11. Click **Save and Close**.
12. Click the `AccountKey` link, then set the following options on the **Edit UI Properties** dialog:
    - Set **Required** to **No**.
    - Set **Updatable** to **Yes**.
    - Set **Hidden** to **No**.
13. Click **Save and Close**.
14. Click **Done**.
15. In the **Details Page Layouts** section, click the **Actions** menu, and select **Duplicate**.
    The **Duplicate Layout** dialog appears.
16. Enter a name in the **New Layout Name**.
17. Click **Save and Edit**.
18. In the **Subtabs Region**, click the **Profile** subtab.
19. Click the **Pencil** icon (**Edit**) next to `SummaryEdit Summary Subtab`.
    The **Details Layout Edit Summary** page appears.
20. From the **Available Fields** column, click the **Account Key** field that you created in the previous task, then click the arrow to move it to the **Selected Fields** column.
21. Click **Save and Close**.
22. Click the **Account Key** link.
    The **Edit UI Properties** dialog appears.
23. Set **Required** to **Expression**, then click the `xyz` next to the **Required** field.
    The **Advanced Expression** dialog appears.
24. In the **Edit Script** text box, add the following expression:

```
if (SalesProfileType == "ZCA_CUSTOMER") {return true} else {return false}
```

> **Note:** This makes the Account Key required if the **Account Type** is **Customer**. Self-service registration only works for **Customer** accounts.

25. Click **OK**.
26. Set **Updatable** to **Yes**.
27. Set **Hidden** to **Expression**, then click the `xyz` next to the **Required** field.
    The **Advanced Expression** dialog appears.
28. In the **Edit Script** text box, add the following expression:

```
if (SalesProfileType != "ZCA_CUSTOMER") {return true} else {return false}
```

> **Note:** This hides the field if it's not a **Customer** account type.

29. Click **OK**.
30. Click **Save and Close**.

Populate the Account Key for Existing Accounts

If you already have existing Customer accounts, you'll need to edit the accounts, and add the required Account Key field.

Note: Make sure to assign unique values to each account.

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Navigate to **Accounts**.
3. Search for, then select the account.

   The **Edit Account** dialog appears.
4. In the **Account Key** field, enter a value that uniquely identifies the Customer account.
5. Click **Save and Close**.
6. Repeat steps 3-5 for all previously existing Customer accounts.

Publish the Sandbox

To ensure that the configuration that you applied in the previous tasks takes effect, you must now publish the sandbox.

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Click **Navigator > Configuration > Sandboxes**.
3. Select the name of the sandbox created in step 2 of the Add the Account Key to the Account Object task.

   The **Sandbox Details** dialog appears.
4. Click **Publish**.

Set a Profile Attribute for the Account Key

The account key is used to uniquely identify an account, and is decided by the value assigned to the SVC_CSS_ACCT_KEY_FIELD profile option.

By default, the SVC_CSS_ACCT_KEY_FIELD profile option is set to the **organizationName** field, which appears as **Name of the Account** in the Oracle Fusion Service User Interface for account management. You must create an attribute to be the account key: the default account key of **organizationName** isn't secure because it can be easily guessed.

You must set the `SVC_CSS_ACCT_KEY_FIELD` profile option to the **API Name** field of the Account Key created in step 7 of the Add the Account Key to the Account Object task in this topic.

Create a Customer Account in Oracle Fusion Service

To set up a customer account in Oracle Fusion Service:

1. Sign in to Oracle Fusion Service.
2. Navigate to the **Service** work area and click **Accounts**.
3. Click **Create Account**.
4. Enter the **Name**.
5. Select **Customer** from the **Type** menu.
6. Specify the account key in the appropriate field.

   Note: The field in which you enter the account key depends on the value assigned to the SVC_CSS_ACCT_KEY_FIELD profile option. The account key should be specified in the attribute that you defined for the account key.
7. Click **Save and Close**.

End User Self-Registration User Account Creation

The Oracle Digital Customer Service Reference Implementation has sample pages that provide the ability for an end user to self-identify and register within the application.

The end user needs the account key to register successfully.

Related Topics

- How do I configure DCS profile options?

Manage Registration Requests

Registration requests are sent to Oracle Fusion Service for users intending to use Digital Customer Service features. Administrators must then determine whether the request should be approved or rejected in the Service work area in Fusion Service.

This topic explains how to accept and reject registration requests in Fusion Service.

> **Note:** Account Administrators can also use the Digital Customer Service application to select multiple pending self-service registration requests to approve or reject.

Approve Registration Requests

This topic describes how to approve registration requests in Oracle Fusion Service.

To approve registration requests:

1. Sign in as a user with a role that includes the Customer Self-Service Administrator duty role.
2. Navigate to the **Service** work area and click **Registration Requests**.

   The **Self-Service Registrations** screen is displayed. By default, a list of pending self-service registrations is displayed.
3. Click the **Actions** list, then select **Approve** .
4. Select one or more pending registration requests that you want to approve.
5. Click the **Approve (# Selected)** button.

   > **Note:** The number sign (#) represents the number of registration requests selected.

   The **Requests to Be Approved** dialog box appears.
6. (Optional) In the **Reason for Approving** text box, enter a reason.
7. Click the **Approve** button.

   A message appears, confirming the number of approved registration requests. The approved requests no longer appear in the pending list.

   > **Note:** When a user registration request is approved, a welcome email is sent to the user with a password reset link.

Reject Registration Requests

This topic describes how to reject registration requests in Oracle Fusion Service.

To reject registration requests:

1. Sign in as a user with a role that includes the Customer Self-Service Administrator duty role.
2. Navigate to the **Service** work area and click **Registration Requests**.

   The **Self-Service Registrations** screen is displayed. By default, a list of pending self-service registrations is displayed.
3. Click the **Actions** list, then select **Reject** .
4. Select one or more pending registration requests.
5. Click the **Reject (# Selected)** button.

   > **Note:** The number sign (#) represents the number of registration requests that you have selected.

   The **Requests to Be Rejected** dialog box appears.
6. (Optional) In the **Reason for Rejecting** text box, enter a reason.
7. Click the **Reject** button.

   A message appears, confirming the number of rejected registration requests. The rejected requests no longer appear in the pending list.

How do I manage Self-Service users?

Users can be granted different roles. By default, when a registration request is approved for a user, they are granted the User role. However, administrators can grant or remove roles, depending on the user's intended responsibilities.

> **Note:** The first user to be approved for a customer account is automatically granted the Account Administrator role. There must always be one user with the Account Administrator role for an account.

Use this topic to add and remove roles for self-service users in Oracle Fusion Service. For more information about self-service user roles, see About Digital Customer Service Roles.

To manage self-service roles for an account:

1. Sign in as a user with the Customer Self-Service Administrator role.
2. Navigate to the **Service** work area and click **Self-Service Users**.
3. From the **Self-Service Users** list, select the user you want to modify.

4. In the **User Administration** section, select or deselect one or more of the following roles:
    - **User**
    - **Account Manager**
    - **Account Administrator**

    Note: Removing the User role causes the removal of all privileges. The only way to restore the privileges is to submit a new registration request. Only a user with the SVC_DELETE_LAST_ACCOUNT_ADMIN_ROLE_PRIV privilege can delete the user role for the last Account Administrator. Users with the Customer Self-Service Administration role have the SVC_DELETE_LAST_ACCOUNT_ADMIN_ROLE_PRIV privilege by default. The last account administrator can only be deleted using the selfServiceRoles REST API.

5. Click **Save**.

    A dialog appears, confirming the role modifications.

How do I Import Self-Service users?

This topic shows you how to import self-service users for use with your Digital Customer Service application.

Use cases for import include when you're migrating from one service implementation solution to Digital Customer Service, or when you've a new customer account and have been supplied with a list of authorized users.

If you're migrating from another service implementation to Digital Customer Service, you might have some existing reference identifiers that you'll want to retain from the originating service implementation. In this particular case, the following fields might help: AccountPartyNumber from the accounts, ContactPartyNumber from the contacts.

First you must create a contact. Once you've created the contact, you've two choices: you can use the Contact Party ID or Contact Party number. When contacts are created or imported, the contact party ID is automatically assigned. You can explicitly specify the Contact Party number. If you choose not to specify the Contact Party number, a number will be automatically assigned.

Accounts are handled in much the same way. An Account Party ID is automatically assigned, and you can choose to specify an Account Party number or have one automatically assigned.

Finally, there's a primary key for the imported data set. The value of the primary key column is automatically generated in the Oracle Fusion Applications data model.

Here's a high level overview of what you'll be doing when importing self-service users:

- Creating contacts
- Downloading the Self-Service Roles template.
- Preparing the Import file, then importing the users.
- Sending pending LDAP requests.

Note: When using the import process for creating self-service users, there's no other approval step required.

Create Contacts

Your first step is to create the contacts. The contact must have an email address. To create contacts, follow the instructions in the Import Your Contact Data task in the Related Topics list. This might also require you do the Import Your Account Data task, which is also in the Related Topics list, if you haven't already done that yet.

Download the Self-Service Role Template

This section describes how to download the self-service roles template.

The Self-Service roles template contains the following fields:

Field	Usage
AccountPartyId	A required field if the AccountPartyNumber value isn't specified. Don't provide this value for B2C.
AccountPartyNumber	A required field if the AccountPartyId value isn't specified. Don't provide this value for B2C.
ContactPartyId	A required field if the ContactPartyNumber value isn't specified.
ContactPartyNumber	A required field if the ContactPartyId value isn't specified.
LoginId	An optional field. If a value isn't specified its assigned the value of a Contact attribute specified by the SVC_CSS_IMP_SIGN_IN_ATTR_NAME profile option. The default value for this profile option is Email Address.
RelationshipTypeCd	A required field. The following values can be assigned: • **ORA_CSS_USER**. This corresponds to the Digital Customer Service User role. • **ORA_CSS_ACC_MGR**. This corresponds to the Digital Customer Service Account Manager role. • **ORA_CSS_ACC_ADMIN**. This corresponds to the Digital Customer Service Account Administrator role.
RequestTypeCd	A required field for B2C. An optional field for Fusion. The following values can be assigned: • **ORA_CSS_REQ_TYPE_CONTACT**. This Fusion Service value corresponds to the Fusion Contact. • **ORA_CSS_REQ_TYPE_CONSUMER**. This B2C Service value that corresponds to the B2C Consumer.
RoleId	An optional field. You can remove from the first row of the file.

To download the Self-Service roles template file:

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. From the Navigator menu, expand **Tools**, and then select **Import Management**.
3. In the **Manage Imports** work area, click the **Import Objects** tab, and select **Search**.
4. In the Import Object Details page, enter **Self-Service Role** in the Display Name search box.
5. Click **Download** and then save the .zip file to a convenient directory.

Prepare the Import File

This section describes how to prepare the import file for importing self-service users.

> **Note:** When users are imported into multiple accounts the data should be sorted by account party ID. This maximizes the possibility of the one thread handling users of the same account and minimize the possibility of deadlocks occurring due to different threads changing the same account. If all the users belong to the same account, then its recommended that you use Oracle Integration Cloud integration and retry the logic for handling account row lock exception.

To prepare the import file:

1. Find, and open the **Self-Service Role_Templates<date>.zip** file that you saved in the Download the Self-Service Role Template topic.
2. Extract, then open the **SelfServiceRole.csv** file.

3. For each self-service user you plan to import, enter the following data in a dedicated row:

    a. Enter the account information relating to the self-service user in the **AccountPartyId** or **AccountPartyNumber** column.

    > **Note:** The Account ID value isn't needed for B2C Service requests.

    > **Tip:** To quickly find the values for **AccountPartyId** or **AccountPartyNumber** using REST API, run the following command as an administrator:
    >
    > `GET <Oracle-Fusion-Application-Host>/crmRestApi/resources/11.13.18.05/accounts/`

    b. Enter the contact information relating to the self-service user in the **ContactPartyId** or **ContactPartyNumber** column.

    > **Note:** When importing the Contacts, it's possible to provide a unique identifier for each contact in the form of the ContactPartyNumber. This must be unique for each user. So you should provide the same unique numbers for ContactPartyNumber for the user in the Contact import file and the Self-Service Roles import file to match them up.

    c. Enter the ID relating to the self-service user in the **LoginId** column.

    > **Note:** If the `SVC_CSS_IMP_SIGN_IN_ATTR_NAME` profile option is set, then the `LoginId` is optional.

    d. Enter the Digital Customer Service roles to assign to the self-service user in the **RelationshipTypeCd** column. The following values can be assigned:

    - `ORA_CSS_USER`. This corresponds to the Digital Customer Service User role.
    - `ORA_CSS_ACC_MGR`. This corresponds to the Digital Customer Service Account Manager role.
    - `ORA_CSS_ACC_ADMIN`. This corresponds to the Digital Customer Service Account Administrator role.

    > **Note:** The ORA_CSS_ACC_MGR and ORA_CSS_ACC_ADMIN roles by themselves can be only used if the self-service user already exists. If the import is going to create a new self-service user the ORA_CSS_ACC_MGR and ORA_CSS_ACC_ADMIN roles must be combined with ORA_CSS_USE role. To assign multiple roles to a user concatenate them using the `&` character. For example: ORA_CSS_USER&ORA_CSS_ACC_ADMIN.

    e. Enter the Digital Customer Service roles to assign to the self-service user in the **RequestTypeCd** column. If a value isn't supplied the default is `ORA_CSS_REQ_TYPE_CONTACT`.

    The following values can be assigned:

    - `ORA_CSS_REQ_TYPE_CONTACT`. This value is for Fusion Service instances and corresponds to the Fusion Contact.
    - `ORA_CSS_REQ_TYPE_CONSUMER`. This value is for B2C Service instances and corresponds to the B2C Consumer. B2C is general public.

4. Repeat step 4, on a dedicated row for each more self-service user you want to import.

    > **Note:** For each `AccountPartyId` or `AccountPartyNumber`, at least one user in the import file must have the `ORA_CSS_ACC_ADMIN` role assigned.

5. Save the **SelfServiceRole.csv** file.

6. Include the **SelfServiceRole.csv** in a new **Self-Service Role_Templates<date>.zip** archive, and save it.

> **Tip:** When using a Mac, you must the `zip` utility in terminal to create the **Self-Service Role_Templates<date>.zip** file.

Consider the following sample data in the **SelfServiceRole.csv** file:

Example 1: **Using Party IDs for import**

```
AccountPartyId,ContactPartyId,RelationshipTypeCd
300100110957452,300100156316610,ORA_CSS_User & ORA_CSS_ACC_ADMIN
```

Example 2: Using Party numbers for import

```
AccountPartyNumber,ContactPartyNumber,RelationshipTypeCd
CDRM_67617,CDRM_743628,ORA_CSS_USER
CDRM_67617,CDRM_743711,ORA_CSS_USER&ORA_CSS_ACC_ADMIN
CDRM_67617,CDRM_743651,ORA_CSS_USER&ORA_CSS_ACC_MGR
```

Example 3: **The Login ID explicitly used**

```
AccountPartyNum,ContactPartyNum,LoginId,RelationshipTypeCd
CDRM_67617,CDRM_743628,Mary.Smith,ORA_CSS_USER
CDRM_67617,CDRM_743711,John.Rogers,ORA_CSS_USER&ORA_CSS_ACC_ADMIN
CDRM_67617,CDRM_743651,Pat.Williams,ORA_CSS_USER&ORA_CSS_ACC_MGR
```

Example 4: **B2C**

```
ContactPartyNum,RelationshipTypeCd, RequestTypeCd
CDRM_943831,ORA_CSS_USER, ORA_CSS_REQ_TYPE_CONSUMER
CDRM_943832,ORA_CSS_USER, ORA_CSS_REQ_TYPE_CONSUMER
```

Import the Self-Service Users

Before proceeding with the instructions in this section, you must have completed the steps in the previous sections:

- Downloading the Self-Service Roles Template
- Preparing the Import File

> **Note:** For best performance set the value of profile option `ZCA_STANDARD_IMPORT_COMMIT_SIZE` to 1. If `SVC_CSS_USE_FA_AS_IDP` is enabled (set to TRUE) you must set the `ZCA_STANDARD_IMPORT_COMMIT_SIZE` value to 1. This must be done for the following reasons. The default value for commit size is 100. This value also influences the `retry` logic built into the import framework. An error in any one import row will cause the entire batch to roll back. For self-service import when IDCS is the IDP, the impact of rollback is severe because many users determined by the commit size have been created and needs to be rolled back. With `retry` the impact is even more severe because there could be more than 1 rollback for the same batch causing the import process to slow down considerably.

To import the self-service users:

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. From the Navigator menu, expand **Tools**, and then select **Import Management**.
3. In the **Manage Imports** work area, click the **Create Import Activity** button.

4. In the **Create Import Activity** workspace, enter the following import options:
    a. In the required **Name** field, enter a name for the import activity.
    b. In the optional **Description** field, enter a description of the import activity.
    c. Click the **Object** drop-down list, and search for **Self-Service Role**.
    d. Click the File Name **Browse** button and find the `SelfServiceRole.csv` file you downloaded.
5. Click **Next**.
6. Verify the mappings are as expected. If necessary, make your corrections.
7. Click **Validate Data**.
8. Click **Next**.

    The **Create Import Activity: Review and Submit** screen is displayed.
9. Click the **Submit** button.

    Your import job is listed in the **Manage Import Activities** list. The **Status** value of your job will change multiple times during processing, until it's **Completed** or **Completed with Errors**.
10. From the **Manage Imports** workspace, click the **My Completed Imports** tile and select the link for your import from the list and review the import details.
11. (Optional) Review errors. If the Status value is **Completed with Errors**, some users might not have been imported. Follow these steps to review the errors:
    a. On the **Import Status** workspace, click the **Actions** menu and select Generate Diagnostics.
    b. Download the generated `zip` file, to review the diagnostic messages .

Send Pending LDAP Requests

Perform this task after successfully importing the self-service user roles.

> **Note:** This topic is only required only if SVC_CSS_USE_FA_AS_IDP profile option is set to TRUE.

With the import of the self-service roles to associate with the contact records, the final step to enable a self-service user account is to create the user account in the Oracle Fusion Service identity management system. This is in turn synchronized with Oracle Identity Cloud Service, allowing users to sign in. To create the user account in the Oracle Fusion Service identity management system, the **Send Pending LDAP Requests** job needs to be run.

To send the pending LDAP requests:

1. Sign in to Oracle Fusion Service as an administrator or a setup user.
2. Navigate to the **Scheduled Processes** work area.
3. Click **Schedule New Process**.

    The **Schedule New Process** dialog box appears.
4. Select the **Job** option.
5. Click the **Name** menu, then select **Search**.
6. Enter the following string in the **Name** text box, then click **Search**:

    ```
    Send Pending LDAP Requests
    ```
7. Click **Send Pending LDAP Requests**, then click **OK**.
8. Click **OK** on the**Schedule New Process** dialog box.

    The **Process Details** dialog box appears.
9. Click **Submit**.

A **Confirmation** dialog is displayed.

10. Click **OK**.

    Monitor the job. When it's complete, the newly imported self-service users are created in LDAP.

Removing Self-Service users through import

You can remove Self-Service users data using Self-Service Role import.

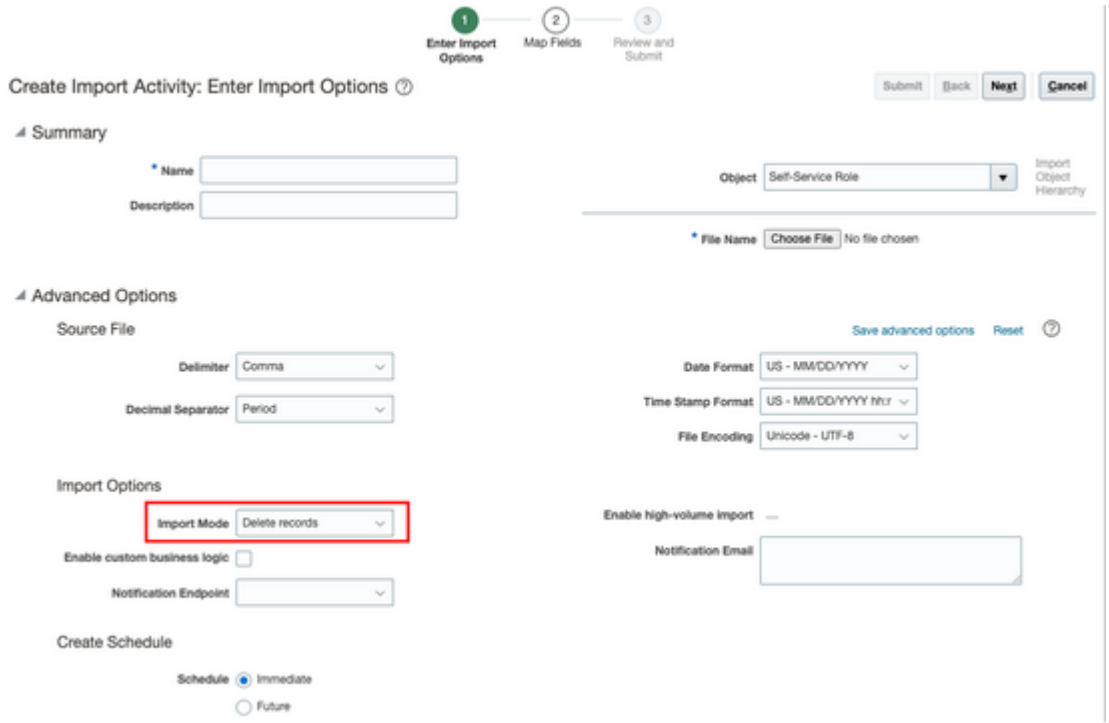
Removing Self-Service User data will not remove the following:

- Any contacts associated with the Self-Service user.
- The user account in the Identity Management System.

First you prepare the data file for the import process.

1. Get the role ID using the the **selfServiceUsers** REST API or **selfServiceRoles** REST API.

   To use to **selfServiceRoles** REST API to get the Role ID of the user you can query by **ContactPartyId** and **RelationshipTypeCd** set to ORA_CSS_USER.
2. When you start the import make sure the **Import Mode** is set to **Delete Records** as shown in the following example:



Self-Service Contact or Account Merge

Fusion Service uses Oracle Customer Data Management to cleanse contact and account records.

Merge actions eliminated duplicate records by creating a victim and survivor record. The survivor record serves as the master record. Digital Customer Service has several rules which are used in Fusion Service to prevent merge operations on victim records.

Here's a list of potential error messages along with resolutions.

Error Message Name and Message Text	Description	How You Resolve the Error
ZCH_GUID_ISSUE This merge request can't be processed because one of the duplicate records is a valid user.	This error message is displayed when one of the duplicates in a merge request is a valid user.	Delete the victim Self-Service user using Self-Service Users UI. If Fusion Service is your Identity Provider do the following: • Use the Security Console to delete the user. • Then, run the Synchronize User GUID job. If Oracle Identity Cloud Service is your Identity Provider do the following: • Delete the user in Identity Cloud Service Administrator Console.
SVC_CSS_CNT_MRG_PENDING_REG This merge request can't be processed because one of the duplicate contact records has pending self-service registration requests.	This message is displayed when two accounts are being merged and the victim record has pending registration requests.	Use the Registration Requests UI to reject the pending registration request of the victim contact.
SVC_CSS_ACT_MRG_PENDING_REG This merge request can't be processed because 1 of the duplicate records has pending self-service registration requests.	This message is displayed when 2 accounts are being merged and the victim record has pending registration requests.	Use the Registration Requests UI to reject the pending registration request of contacts from the victim account.
SVC_CSS_ACT_MRG_USERS This merge request can't be processed because 1 of the duplicate records has existing self-service users.	This message is displayed when 2 accounts are being merged and the victim record has existing self-service users.	Currently there's no resolution (one will come in a future release).

8 Knowledge User Groups

Create a Knowledge User Group

Create a knowledge user group using this task.

1. Sign in to Fusion Service as an administrator or setup user.
2. In the Setup and Maintenance area, go to the following:
    - Offering: Service
    - Functional Area: Knowledge Management
    - Task: Manage Knowledge User Groups
3. From the **Department** drop-down list, select **Service**.
4. Click **Create new user group** (**+**) to add a new user group.

5. Enter a unique name for the user group.
6. Enter a reference key name using only capital letters, numbers and underscore.

    Note: You will need this new knowledge user group's reference key later when you create a data security policy for the new role that you will create.

7. Click **Create**.

For more information on working with user groups, see the Related Topics link.

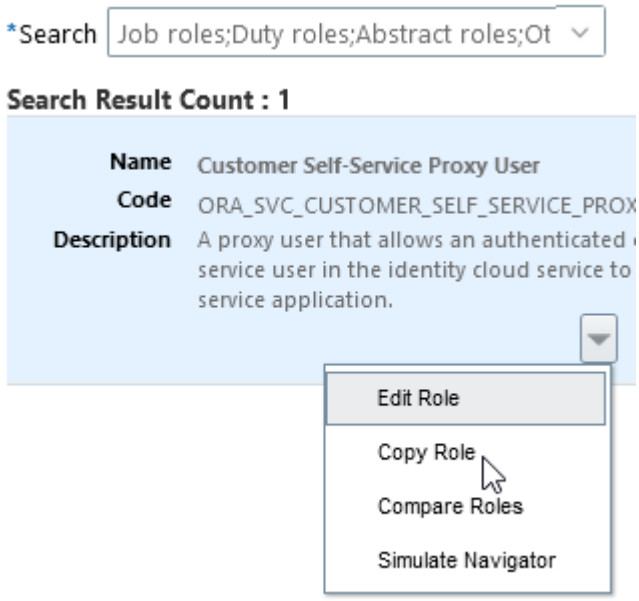
Related Topics

- How do I implement user groups in Knowledge?

Create a Custom Job Role and Assign Two Data Security Policies

In this topic you create a copy of the Customer Self-Service Proxy User role and assign a user group data security policy to the custom role.

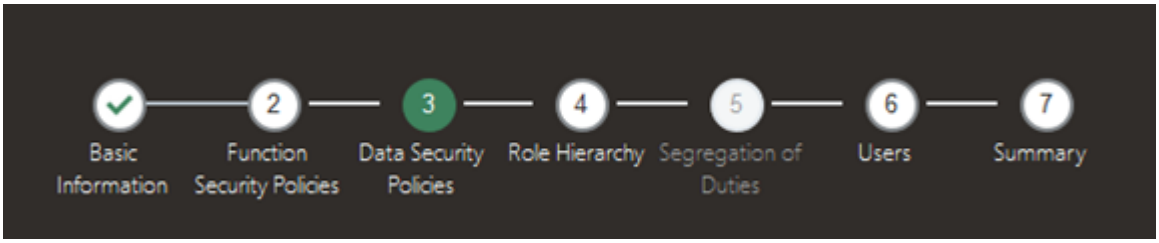
1. In Fusion Service, click Navigator, expand the Tools section and select **Security Console**.
2. Click the **Roles** tab.
3. Search for and select the **Customer Self-Service Proxy User** role. The role code is: ORA_SVC_CUSTOMER_SELF_SERVICE_PROXY_USER_ABSTRACT.
4. In the Search Result field, click the drop-down list and select **Copy Role**.



5. In the **Copy Options** dialog box, select **Copy top role**, then click **Copy Role**.
6. On the Basic Information page, enter a role name, such as **Knowledge User Group Proxy User**, a role code (SVC_KNOWLEDGE_USER_GROUP_PROXY_USER), and optionally a description.

> **Note:** Make a note of the Role Code as you'll need it in last tasks when you create an Identity Cloud Service group.

7. Click Step 3, **Data Security Policies**.



8. Create a data security policy by doing the following:
   a. Click **+ Create Data Security Policy** and change the start date to any date earlier than today's date.
   b. Enter your policy name, such as **Knowledge User Group Proxy User**.
   c. Click the **Database Resource** search icon and search for **Knowledge User groups**, select that database resource, and click **OK** to associate it with the newly created policy.
   d. Click the **Data Set** drop-down list, and choose **Select by key**.
   e. Enter the user group reference key you created in the Create a Knowledge User Group task in the **Select a primary key value** field.
   f. Click the **Actions** drop-down list, and select **Access Content with User Group**.

g. Click **OK**.

9. Now, create a data security policy to grant access to the Service Department by doing the following:
   a. Click **+ Create Data Security Policy** and change the start date to any date earlier than today's date.
   b. Enter a policy name, such as **Knowledge Service Department Policy**.
   c. Search for: **Knowledge Departments**, select the **Database Resource**, then click **OK**. .
   d. Click the **Data Set** drop-down list, and choose **Select by instance set**.
   e. Click the **Condition Name** drop-down list, and select **Access to the service department**.
   f. Click the **Actions** drop-down list, and select **Access Content with Department**.
   g. Click **OK**.
10. Click Step 7, **Summary** and review your work.
    You should see something like the following:
    - Function Security Policies: Added (0), Removed (0)
    - Data Security Policies: Added (2), Removed (0)
    - Role Hierarchy: Added (0), Removed (0)
    - Users: Added (0), Removed (0)
11. Click **Submit and Close**.
12. Back in Security Console, verify the role copy was successful by doing the following:
    a. Click the **Administration** tab.
    b. Click **Role Copy Status** and find the Role Code you entered in the previous step.
    c. Verify the status for your process shows **Complete**.

Related Topics

- How do I implement user groups in Knowledge?

Create a Proxy User and Grant the Job Role

Next, create a proxy user and assign the custom job role you previously created.

1. Sign in to Oracle Fusion Service.
2. Navigate to the Security Console work area and click the **Users** tab.
3. Click **Add User Account**.
4. Assign that user the custom job role you previously created, Enter the values in the required fields, then click **Add Role**.
   The following table shows example values.

Field	Value
Associated Person Type	None.
Account Information	For Active, set to **True**. For Locked, set to **False**.

Field	Value
User Category	Click the drop-down list, and choose: PROXYUSERS.
First Name	Leave blank.
Last Name	Enter: Knowledge User Group Proxy User.
Email	Leave blank.
User Name	Knowledge_User_Group_Proxy_User.
Password	Enter a password and record it for future use.
Confirm Password	Reenter the password.

5. Click the **Add Role** button.
6. In the **Add Role Membership from Role** dialog box, search for and select the custom role you previously created.
7. Select the custom role, then click **Add Role Membership**, and then click **Done**.
8. On the **Add User Account** page, click **Save and Close**.

Add the New Proxy User to the FND_IDP_PROXY_USER_WHITELIST Profile Option

Use this topic to add the newly created proxy user to the FND_IDP_PROXY_USER_WHITELIST profile option

1. In the Setup and Maintenance work area, go to the following:
   - Offering: Service
   - Functional Area: Digital Customer Service
   - Task: Manage Digital Customer Service Profile Options
2. Select the FND_IDP_PROXY_USER_WHITELIST profile option.
3. In the Profile Values workspace verify that **Site** is set as the Profile Value.
4. Add the user name of the proxy user you created in the Create a Proxy User and Grant the Job Role topic.

   > **Note:** The list of values is a comma separated list with no spaces.

5. Click **Save and Close**.

Verify that Identity Cloud Service has a Corresponding Group

Use this task to discover the IDCS URL, and following that, see the related links to create the new IDCS group.

Note: Don't create the group using the IDCS UI, use the Functional Setup Manager task listed at the end of this topic. If , however, you do create a group in IDCS, make sure that the external ID of the IDCS group you create matches the code name of the Fusion Service custom job role you previously created.

1. If the Knowledge Role created does not exist in IDCS, use the Functional Setup Manager in Fusion to create the group.

   To do this, refer to *Create groups in Fusion Service*.

   Note: The external ID of the IDCS group must match the code of the Fusion Service custom job role you created previously.

2. Assign each Self Service user who needs to see knowledge articles associated with the user group.

Modify Existing Self-Service Optimization Configuration Data

Use these tasks to modify your existing configuration data for Self-Service Optimization.

Modify the Priority of the Existing Knowledge Proxies

Use this task to modify the priority of the existing knowledge proxies.

1. In the Setup and Maintenance work area, go to the following:
   - Offering: Service
   - Functional Area: Digital Customer Service
   - Task: Manage Proxy User Configuration Data
2. In the URL pattern field enter the following: **srt**.
3. Click **Search** to view the knowledge search proxies.
4. In the Search results, locate the row for `/srt/api/.*/search/question` for Customer Self-Service User, and then do the following:
   a.
   b. Click the link in the **URL Pattern** column.
   c. Change the value of the **Priority** field to a value greater than 1, such as 5.
   d. Click **Save and Close**.
5. Locate the `/srt/api/.*/search/answer` row and repeat the actions from the previous step.
6. In the **URL pattern** field, enter content and click **Search** to view the knowledge content proxies.
7. Locate the `/km/api/.*/content` for Customer Self-Service User, and then do the following:
   a. Click the link in the **URL Pattern** column.
   b. Change the value of the **Priority** field to a value greater than 1.
   c. Click **Save and Close**.
8. Click **Done**.

Add a New Knowledge Question Proxy for the New Proxy User

Next you add a new knowledge question proxy for the new proxy user you have already created.

1. From the **Manage Proxy User Configuration Data** page, create a new record by clicking the **Create** icon (**+**) in the Search Results work area.
2. In the **Create Proxy User Configuration Page**, enter the following values shown in the following table:

Field	Enter This Value
URL Pattern	/srt/api/.*/search/question
HTTP Methods	Select the following: - Options - Post
IdP User Roles	Select the name of the IDCS group you previously created in the Create a New Group in Identity Cloud Service task.
Priority	1 You can enter any value, but remember that the proxies are executed in priority order where 1 is the highest priority.
Proxy User Key	Create the Proxy Key by doing the following: a. Click the drop down list, and select **New Key**. **Note:** The key name must begin with **PUK#_**. b. In the **Proxy User Name** field, enter the name of the proxy user you previously created in the Create a Proxy Users and Grant the Job Role task. c. Enter and reenter the password for the proxy user. d. Select the **Allow Identity Propagation** check box.

3. Click **Save and Close**.

Note: Proxy user configuration data is cached so changes won't be reflected until the cache expires. To show your changes sooner, force the cache to be cleared by changing the value of the profile option SVC_CSS_PUDS_CACHE_DURATION. You can make the value 1 second higher or lower.

Add a New Knowledge Answer Proxy for the New Proxy User

Now you add a new knowledge answer proxy for the new Proxy User.

1. From the **Manage Proxy User Configuration Data** page, create a new record by clicking the **Create** icon (**+**) in the Search Results work area.
2. In the **Create Proxy User Configuration Page**, enter the following values shown in the following table:

Field	Enter This Value
URL Pattern	/srt/api/.*/search/answer
HTTP Methods	○ Options ○ Post
IdP User Roles	Select the name of the IDCS group you previously created.
Priority	1 You can enter any value, but remember that the proxies are executed in priority order where 1 is the highest priority.
Proxy User Key	Click the drop down list, and select the proxy user key you created previously. Note that the values of the proxy user name and password auto populate.
Allow Identity Propagation	Select the check box.

3. Click **Save and Close**.
4. Click **Done**.

Add a New Knowledge Content Proxy for the New Proxy User

Now add a new knowledge content proxy for the new Proxy User.

1. From the **Manage Proxy User Configuration Data** page, create a new record by clicking the **Create** icon (**+**) in the Search Results work area.
2. In the **Create Proxy User Configuration Page**, enter the following values shown in the following table:

Field	Enter This Value
URL Pattern	/km/api/.*/content
HTTP Methods	○ Get

Field	Enter This Value
	○ Options
IdP User Roles	Select the name of the IDCS group you previously created.
Priority	1 You can enter any value, but remember that the proxies are executed in priority order where 1 is the highest priority.
Proxy User Key	Click the drop down list, and select the proxy user key you created previously. Note that the values of the proxy user name and password auto populate.
Allow Identity Propagation	Select the check box.

Clear the Proxy User Configuration Data Cache

Now use the following task to clear the cache.

1. In the Setup and Maintenance work area, go to the following:
   - Offering: Service
   - Functional Area: Digital Customer Service
   - Task: Manage Digital Customer Service Profile Options
2. Select the **SVC_CSS_PUDS_CACHE_DURATION** profile option.
3. In the **Profile Values** work area, change the profile value by 1 second.
4. Click **Save and Close**.

Create Knowledge Articles in Fusion Service and Assign them to a User Group

Here's how you create knowledge articles in Fusion Service authoring and assign them to the appropriate user group.

1. In Fusion Service, navigate to **Knowledge Authoring**.
2. Select **Create Article**.
3. Select the appropriate content type: Solution, FAQ.
4. Enter the requested information and click **Next**.
5. Select the user groups that apply.
6. Click **Create**.
7. Publish the knowledge articles.

8. Run the knowledge content processing job by doing the following:
    a. In Setup and Maintenance, choose the following:
        - Offering: Service
        - Functional Area: Knowledge Management
        - Task: Manage Knowledge Search Dictionary
    b. Select the **Content Processing** tab.
    c. Select **Run Incremental Content Processing**.
    d. To see if content indexing is complete, click **Refresh**.

9 Components

Overview of Components

This topic describes the Digital Customer Service components that are available when developing your application in the Oracle Visual Builder.

There are two types of components:

- **Oracle JavaScript Extension Toolkit Composite Component Architecture components.** These components are reusable pieces of user interface code that you can embed as custom HTML elements in the Visual Builder Application.
- **Oracle Visual Builder pattern components.** These components use templates of existing available components and HTML elements and additional Oracle Visual Builder-related bindings such as page variables, listeners and action chains.

Oracle Visual Builder pattern components aren't accessible as a component whose attributes you can inspect. Once included on a page, the Oracle Visual Builder pattern components are a collection of other HTML components with no outer component. Whereas the Standard Oracle JavaScript Extension Toolkit Oracle JET Composite Component Architecture components have attributes that can be inspected.

Oracle Visual Builder pattern components that are a part of Digital Customer Service are:

- Asset Detail
- Asset List
- Asset Register
- Change Password Form
- Forgot Password Form
- Knowledge Article
- Knowledge Article Facets
- Knowledge Article List
- Knowledge Popular Articles
- Knowledge Search
- Reset Password Form
- Service Request Creator
- Service Request Data
- Service Request File Attachment
- Service Request List
- Service Request Message List
- Service Request Message Creator
- Service Request URL Attachment List

- Sign In
- Sign In Form
- Sign Out
- Sign Up
- User Registration Request Management
- User Registration Requests List
- User Roles Management
- User Roles List
- Visual Navigator
- Work Order Data
- Work Order List

Standard Oracle JavaScript Extension Toolkit Oracle JET Composite Component Architecture components that are a part of Digital Customer Service are:

- Chat
- Cobrowse
- Category Selector
- Product Selector
- Linked Text
- Download Attachment

What are the Installed Base Asset components?

If you're an Oracle Cloud customer that uses 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 your service request and work order processes.

Your users can register products as installed base assets using the asset-register flow in the Digital Customer Service reference implementation. Users access the flow by clicking the Register Product button on the Registered Products view. You can only register products that have been configured to allow tracking as assets.

Note: Assets and product registrations are Oracle Fusion Installed Base Assets.

Account managers can view and manage all assets associated with the account that they manage. Non-account manager users can only access and manage assets that they have registered themselves.

Users can do the following with registered products:

- View a list of product assets
- Register a product asset
- View the details of a product asset
- Update the product asset description

- See knowledge articles associated with a product asset
- See service requests logged against a product asset
- Create a new service request against a product asset from the product asset details page

Users access the list of registered products by clicking the Settings and Actions menu, and selecting the **Registered Products** option from the drop-down list.

The following table describes components specific to asset registration.

Component Name	Description
Asset List	Lists the Installed Base Assets of Digital Customer Service self-service users. Users can register products as installed base assets using the asset-register flow in the Digital Customer Service Reference Implementation, by clicking the Register Product button on the Registered Products flow. Only products that have been configured to be tracked as assets can be registered.
Asset Register	Enables a self-service user to register an Installed Base Asset in the Digital Customer Service application.
Asset Detail	Enables a self-service user to view details and update the description of the registration of an Installed Base Asset in the Digital Customer Service application.

How Digital Customer Service User Roles are Used by the Installed Base Asset Components

Your users can register products as installed base assets using the asset-register flow in the Digital Customer Service reference implementation. Users access the flow by clicking the Register Product button on the Registered Products view. You can only register products that have been configured to allow tracking as assets.

Here's a list of the roles required for registering products and what each role can do:

Role	Definition
Account Manager	Can view and manage all assets associated with the account being managed.
Non-Account Manager User	Can only access and manage assets that they have registered.
B2C User	Can register and manage assets.

Interaction Methods Components

This topic is lists and describes interaction methods components.

The following table describes components specific to interaction methods.

Component Name	Description
Chat	Enables users to initiate a chat request and exchange messages with an agent.
Cobrowse	Enables users to initiate a Cobrowse session. Cobrowse is launched using the supplied launcherUrl attribute value and the JET language by loading the launcher.js file into the application. The Cobrowse component is by default included on the shell page of the Digital Customer Service Reference Implementation template. For more information, refer to the Configure Cobrowse topic in the Related Topics.

Related Topics

- Configure Cobrowse

Product and Category Components

This topic lists and describes product and category components.

The following table describes components specific to products and categories.

Component Name	Description
Category Selector	Enables users to select a category, for example to filter the search results. This component works in conjunction with Knowledge Search, Chat, Create Service Request and Edit Service Request.
Product Selector	Enables users to select a product, for example to filter the search results. This component works in conjunction with the Knowledge Search, Chat, Create Service Request and Edit Service Request.

Knowledge Management Components

This topic lists and describes knowledge management components.

The following table describes components specific to knowledge management components.

Component Name	Description
Knowledge Article	Enables you to display a knowledge article when the component's variable `kmContentId` is configured with a valid Knowledge Article ID.
Knowledge Article List	Displays a list of knowledge articles associated with a supplied search term and optionally selected product and category filters.

Component Name	Description
Knowledge Popular Articles List	Displays Popular Knowledge Articles. This component provides a user interface component that displays popular articles that are measured by a ranking score. This ranking score is derived from the strength of the knowledge article viewed within a time lapse mechanism. The following variables are associated with this component: • **`kmPopularArticlesList`** - The Service Data Provider for the component which maps the Service Data Provider to the following custom fetch action chain: **`kmPopularArticlesFetchAction`**. • **`kmPopularArticlesFilter`** - The product category filter that adds context to the query. The Action chain associated with the Popular Articles Fetch has a description with details on how **`kmPopularArticlesFetchAction`** can be used. **`kmPopularArticlesFetchAction`** queries the most popular items back from the Knowledge Management content REST API. The number of results can be limited by updating the **`kmPopularArticlesLimit`** NUMBER page variable. • **`kmPopularArticlesLimit`** - Controls the maximum number of articles to display. The default value is 5. • **`kmPopularArticlesResultsCache`** - Represents the object used to scope the caching of the popular articles, and prevents an article from being called more frequently than appropriate.
Knowledge Search	Displays a list of knowledge articles associated with a supplied search term and optionally selected product and category filters. This component can used in collaboration with the Knowledge Article List and Knowledge Article Facets components.
Knowledge Article Facets	Enables you to filter a list of knowledge articles resulting from a knowledge search. Types of facets include: Product, Category, Collection, and Document Type.

Service Requests Components

This topic lists and describes service requests components.

The following table describes components specific to Service Requests. It also lists the restrictions associated with the component and any additional configuration required.

Component Name	Description
Download Attachment	Enables the Service Request Attachment List to support downloading of the listed attachments.
Linked Text	Replaces a text field with references to service requests with links to the details page for the service request, and creates links to Knowledge Management articles. For example, "SR1234567890" in the **Field** value would be replaced with a link to the details page for service request number 1234567890.
Service Request Creator	Displays the service request creation form and the functionality required to allow users to create Service Requests.

Component Name	Description
Service Request Data	Displays the summary details of a service request. For the Service Request Data component to render a valid service request, it must be mapped to the input page **`serviceRequestNumber`** parameter. This parameter is a required parameter and has to be wired to a valid service request number by the calling page navigation.
Service Request File Attachment	Lists service request file attachments and allows the user to edit the description or add new file attachments. For the Service Request File Attachment component to render a valid service request, it must be mapped to the input page **`serviceRequestNumber`** parameter. This parameter is a required parameter and has to be wired to a valid service request number by the calling page navigation.
Service Request List	Displays a list of service requests to the account user in a preconfigured Oracle Visual Builder list view. The presentation of a row can be edited.
Service Request Message Creator	Provides functionality to allow new messages to be added to a service request.
Service Request Message List	Displays the messages that are associated with a service request.
Service Request URL Attachment List	Lists service request URL attachments and allows the user to edit the description or add new URL attachments. For the Service Request URL component to render a valid service request, it must be mapped to the input page **`serviceRequestNumber`** parameter. This parameter is a required parameter and has to be wired to a valid service request number by the calling page navigation.

What are the work order components?

This topic lists and describes work order components.

The following table describes components specific to Work Orders. It also lists the restrictions associated with the component and any additional configuration required.

> **Note:** To use work orders in your Digital Customer Service application, you must first complete the setup of either general work orders or Oracle Field Service work orders with Fusion Service. For more information, refer to the following guides:
>
> - Integrating Fusion Service with Field Service
> - Implementing Fusion Service

Component Name	Description
Work Order List	Displays a list of work orders to the account user in a preconfigured Oracle Visual Builder list view. The presentation of a row can be edited. Provides the capability to search for a work order and sort the list of work orders.
Work Order Data	Displays the summary details of a work order. Provides the capability to update contact information, add a message for the technician, reschedule a work order and cancel a work order. It also provides information about the technician and displays the technician's location. **Note:** Self-Service users can only reschedule Oracle Field Service work orders, not generic work orders.

Sign In and Sign Out Components

This topic lists and describes sign in and sign out components.

The following table describes components specific to Sign In and Sign Out.

Component Name	Description
Sign In	Enables users to sign in.
Sign Out	Enables users to sign out.

User Components

This topic lists and describes user components.

The following table describes components specific to users.

Component Name	Description
Sign Up	Allows anonymous users to register to become self-service users.
User Registration Request Management	Enables Oracle Digital Customer self-service administrator users to manage user registrations. The component shows the selected user's name, email, account information, status and a reason associated with the approval or rejection. For the User Registration Request Management component to render a user's registration details, the user's ID must be mapped to the component's page variable named selfServiceRegistrationId.

Component Name	Description
	This parameter is a required parameter. It must be configured by the calling page navigation -- typically from an instance of the User Registrations Requests List -- in that calling page's userRegistrationsNavigateToEditAction action's navigateToEdit step. When in read-only mode, the approver and approved date will be shown.
User Registration Requests List	Lists the Digital Customer Service self-service registrations. The listing can be sorted by last update date, email, name, role or account in either descending or ascending order. The list displays the **username**, email, account, registration status and last update date.
User Roles List	Lists the Digital Customer Service self-service users and their roles. The list can be sorted by last update date, email, name, role or account in either descending or ascending order. The list displays the **username**, email, account, roles and last update date. The rows have a trash can icon, which can be used to remove the user.
User Roles Management	Allows the Digital Customer Service self-service administrator users to view, add and remove user roles. The component shows the selected user's name, email, account information and their assigned roles and details on who made the last update and when. For the User Roles Management component to render a user's roles, the user's ID must be mapped to the component's page variable named userRolesUsersId. This parameter is a required parameter. It must be configured by the calling page navigation -- typically from an instance of the User Roles List -- in that calling page's UserRolesNavigateToEditAction action's navigateToEdit step. The remove user button removes the user.
Change Password Form	Component that creates a form that allows a user to change their password.
Forgot Password Form	Component that creates a form that allows users who have forgotten their password to initiate the reset password process.
Reset Password Form	Component that creates a form that allows users who have forgotten their password to complete the reset password process.
Sign-In Form	Component that creates a form that allows IDCS users to log in to the DCS application.

Integrate Composite Component Architecture Components

You can integrate non-Digital Customer Service Oracle JET Composite Component Architecture (CCA) components with your application.

For more information about this type of integration, refer to the Interviews for Digital Customer Service topic in the Oracle Intelligent Advisor documentation in the Related Links.

Related Topics

- Add an interview to Digital Customer Service

View Information about Components

This topic describes how to view additional information about components.

You can find more information about component attributes directly in Oracle Visual Builder.

To view more information about a specific component:

1. Navigate to the Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click **Web Apps**.
4. Expand `dcs`, then expand `flows`.
5. Select a page where the component appears.
6. In the **Page Structure** panel, click the component.
7. Click the **Design** tab in the **<Component> Selector** inspector.
8. All of the attributes specific to the selected component appear in the **<Component> Selector** section of the inspector, in the **All** tab.
9. Hover over the name of the attribute, then hover over the question mark icon to reveal one or more of the following fields relating to the attribute:
    - **Type**
    - **Value**
    - **Supported Values**
    - **Description**

10 Life Cycle Management

Determine Component Versions

Digital Customer Service provides reusable components to your application through a component catalog service integrated into Oracle Visual Builder. These components may be updated periodically to provide bug fixes or additional features.

This topic describes how to determine your Digital Customer Service component versions in Oracle Visual Builder.

To determine your Digital Customer Service application component versions:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click the **Puzzle Piece** icon (Components).

   The **Components** list appears.
4. Click the **Installed** tab.

   The names of all of your installed components are listed, including version numbers.

Determine Template Version

This topic describes how to determine the version of the Digital Customer Service template your application was based on, in Oracle Visual Builder.

Digital Customer Service provides templates for you to create your application. More templates might be provided with each version, and in some situations, you might want to know the version of the template that your application was based on.

To determine the version of your template:

1. Sign in to Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click **Web Applications**.
4. In **Web Apps**, click `dcs`.
5. Click the **JS** icon (Functions).

   The functions appear.
6. The first line of the functions will look similar to the following:

   ```
   /* Oracle Digital Customer Service Template 19C 2.3.1: master a71f210 2019-06-06T02:07:28.700Z */
   ```

   In this example, the **19C 2.3.1** indicates the version of the template that you're using.

Updates and Upgrades

Application Upgrade Dependencies

Periodically Oracle updates the underlying version of Visual Builder. When an upgrade occurs, the Visual Builder Runtime and Oracle JET versions are also updated.

If a new Digital Customer Service application is created, it's automatically set to use the latest version of the runtime dependencies. Although Oracle recommends you always run the latest versions for your existing Digital Customer Service applications, it's for you to decide when the appropriate time is for you to upgrade the components.

Before you start any upgrade always backup your Digital Customer Service application first. You can do this in any one of the following ways:

- Export the application and save the archive file.
- Create a new version of the application.
- Duplicate the application to create a new application.
- Push the changes to a Git repository.

Once the application has been upgraded you must stage and publish it again.

Upgrade Visual Builder Runtime and JET Version

Periodically Oracle updates the underlying version of Visual Builder. When an upgrade occurs, the Visual Builder Runtime and Oracle JET versions are also updated.

If a new Digital Customer Service application is created, it's automatically set to use the latest version of the runtime dependencies. Although Oracle recommends you always run the latest versions for your existing Digital Customer Service applications, it's for you to decide when the appropriate time is for you to upgrade the components.

Before you start any upgrade always backup your Digital Customer Service application first. You can do this in any one of the following ways:

- Export the application and save the archive file.
- Create a new version of the application.
- Duplicate the application to create a new application.
- Push the changes to a Git repository.

Once the application has been upgraded you must stage and publish it again.

Updates to the oj-odcs JET Pack

When a newer version of the oj-odcs JET components is available, you can install it using the Updates tab in the Components pane.

You'll know an update is available when you see a notification in your browser window or a badge over the Components icon in the Navigator.

To update a component from the Component Exchange, do the following:

1. Open the **Components** tab in the Navigator.
2. Open the **Updates** tab in the Components tab.
3. Click **Update All** to install all updates available for installed components.

   To update an individual component, click the component's name to open its detail page, then click the **Update** button.

If the installed component isn't compatible with the JET version in your Visual Builder instance, you'll see a notice to that effect. For more information, see the Update a Component from the Component Exchange topic.

Manage Your Instance Patching Window

Visual Builder's functional updates are typically provided in two windows which are typically two weeks apart. With Visual Builder you can choose the update window for your instance.

It's recommended that you update non-production instances in the first window (Window 1) and your production instance in the second window (Window 2). Using this upgrade method you can test your applications before the update is applied to your production environment.

Here's a quick overview of the process:

1. Open the Visual Builder instance's Tenant Settings editor.
2. From the Patch drop down list, select the Window option.

   There are only two options: Window 1 and Window 2. Window 1 is the default option. For a more detailed look, refer to the Related Topics.

Related Topics

- What's New in Oracle Visual Builder
- Choose Your Instance's Update Window

Promote Your Application to Another Instance

Promote Your Digital Customer Service Application to Other Instances

Here's the recommended approach for managing the lifecycle of your application from the development phase up through the publishing of your application.

Your Fusion Service instance is provisioned with a Non-Production and Production environment, with some customers also having additional non-production instances. With each of your Fusion Service instances a Visual Builder instance is provisioned. This means there is a distinct Visual Builder associated with your Non-Production and Production environments respectively. The Non-Production instance will also include a Visual Builder Studio instance, enabling you to use a Git repository and a dedicated development environment.

> **Note:** When promoting your application to the production environment consider following the *Performance Tuning Checklist*. In these steps you'll be using the Digital Customer Service application and Visual Builder.

These multiple instances allow development to occur in safe, independent environments, without affecting any production applications. We recommend that development is carried out in your Non-Production environment and promoted to Production only when ready. Additionally, if you have Development environments, these can be used to further segregate the phases of development and provide restricted, controlled access before promoting to the next.

When your Digital Customer Service application is ready to be promoted from the Non-Production or Development Visual Builder environment to the Production environment there are two options available. These options are described in the following topics.

> **Note:** You should not promote to a new environment until you've performed all the steps listed in Chapter 3, Mandatory Setup Tasks.

Promote an Application Using Export and Import

You can export your Digital Customer Service application as an archive to your local system, and then import it to create a new visual application. You can use the import and export mechanism to share application sources and to move applications between instances.

When you export the application, you can choose whether you want the archive to include the data contained in the application's business objects. Some information, such as credentials for external REST end points, is removed when you export an application. This information must be provided after the archive is imported. The application archive can be imported to the required instance via the import tool. You can also replace an existing visual application's source files with content from an archived file using the Import command located in the Application menu.

Once you're done, you then set up the user role mappings.

Finally, stage and publish the application. If you're using a Vanity URL, then make sure you've followed the steps from *How do I set up vanity URLs?*.

For instructions on the export and import framework, refer to *Import and Export Visual Applications*.

Perform Required Prerequisite Steps Before Promoting the Application Using Visual Builder Studio

If you're not already using Visual Builder Studio, then there a few steps required before you can start developing your new Digital Customer Service application.

- Set up the required IDCS roles.
- Get access to the Visual Builder instance.
- Create a project for the Visual Application using the Digital Customer Service template.
- Set up the development project

For instructions on these steps, refer to *Set Up VB Studio for Developing Visual Applications*.

Once you create a Digital Customer Service project using the Visual Application template, several artifacts are created for you:

- A Git repository that contains the visual application's source code.
- A Development environment that points to the Visual Builder development instance.
- Optionally, a private workspace to edit the visual application in the Visual Builder Studio Designer.

Once your Digital Customer Service application is set up in Visual Builder Studio, it's possible to promote it to your production Visual Builder instance, or to any other instance. You can do this by using the Import Application from Visual Builder Studio Git option in Visual Builder.

Promote the Application Using a Git Repository

If your application is stored in a Git repository, then you can import it to the Production environment.

To do this, you first have to make the existing Git repository accessible to the production environment by doing the following:

1. Log into Visual Builder.
2. Select **Import** and then **Import Application from Visual Builder Studio Git**.
3. Select **Add Credentials**.
4. Enter the Git repository URL using the following format: `https://host/organization`.
5. Enter the user name and password of the Git user, and then click **Save Credentials**.
6. From the **Project Selection** drop down list, select your project.
7. From the **Repository Selection** drop down list, select your repository.
8. From the **Branch Selection** drop down list, select your branch.
9. Give the application a name, an ID and a brief description.
10. Click **Save Configuration.**

Once you're done, you then set up the user role mapping.

Finally, stage and publish the application. If you're using a Vanity URL, then make sure you've followed the steps from the *How do I set up vanity URLs?*.

Related Topics

- Performance Tuning Checklist
- Import and Export Visual Applications
- Set Up VB Studio for Developing Visual Applications

Application Modifications

After the application has been promoted to the new environment it's possible that some modifications may be required. Each environment has its own Identity Cloud Service so if your application or user flow makes calls to IDCS then changes are required.

The types of configurations could include:

- Your own sign in page
- A custom password reset
- Cobrowse setup
- Custom endpoint calls. These include any direct calls made in your application to a service. This only applies if the service address is different in the new environment.

For more information, see the Related Topics links.

Related Topics

- How do I enable my own sign in pages?
- How do I enable my own Forgotten Password page?
- Configure Cobrowse

Promote Your Fusion Service Proxy User Data to Another Instance

Export Configuration Data from the Stage Environment

The proxy user name and password aren't moved by this process. Proxy user configuration data must be configured in the stage environment and moved to the production environment.

Note: You perform this step in your Fusion Service environment.

1. Sign in to Fusion Service as administrator or setup user.
2. In the Setup and Maintenance work area, go to the following:
    - Offering: Service.
    - Functional Area: Digital Customer Service.
3. From the Actions drop down list, select **Export**, then **Create New**.
4. Click **Yes** to dismiss the Warning dialog box.
5. In the **Export Offering Setup Data** work area, click the **Show** drop down list in the **Business Objects** area and select **All.**
6. Make sure **Proxy User Configuration** is selected for export. If you don't want to export profile options, deselect **Export** for the Application Profile Value row.

7. If you deselected the Application Profile Value row, click through the Warning pop up window.
8. Enter a name in the Process Name field and then click Submit.
9. Click **OK** to confirm the export, and then monitor the status of the export.
10. When the status reads **Ready to Export**, click the link.
11. Click **Download File**, and then click **Done**.
12. A compressed file with the name you gave the process will be downloaded locally.

Import Configuration Data Into the Production Environment

Now you import the data into your production environment.

Note: You perform this step in your Fusion Service environment.

1. Sign in to Fusion Service as administrator or setup user.
2. In the Setup and Maintenance work area, go to the following:
    - Offering: Service.
    - Functional Area: Digital Customer Service.
3. From the Actions drop down list, select **Import**, then **Create New**.
4. Click **Yes** to dismiss the Warning dialog box.
5. In the **Import Offering Setup Data** work area, click **Browse** and select the compressed file you downloaded in the previous step.
6. Click **Submit**, and then click **OK** to confirm the import.
7. Click the **Waiting for manual import** link.
8. In the Resume Import Setup Data window, click Yes to resume the import.
9. When the status reads **Ready for data validation**, you can validate it using the Manage Proxy User Configuration task.

11 Migrating to Self-Service Optimization

Overview of Migrating Users to Self-Service Optimization

Self-Service User Optimization makes it possible to use Identity Cloud Service for authentication.

With Identity Cloud Service you have a sign-in page that you can configure, along with email notification and immediate login access after registration. This enables you to scale Self Service Optimization to a much larger number of users than would occur in many B2C scenarios.

During the registration approval process, the SVC_CSS_USE_FA_AS_IDP profile option determines the overall behavior.

Here are the profile option settings:

- If the profile option is set to TRUE, the user account is created in the Identity Store managed by Fusion Service, and then copied over to Identity Cloud Service.
- If the profile option is set to FALSE, the user is created directly in Oracle Identity Cloud Service.

A user that's created in the Fusion Service Identity Store, with the SVC_CSS_USE_FA_AS_IDP profile option set to TRUE, must also be enabled for Self-Service Optimization prior to having access to the Digital Customer Service application. The user migration process enables these user accounts.

Here are some terms to keep in mind during your setup:

- Self-Service Optimization enabled Digital Customer Service application: A Digital Customer Service application that's Self-Service Optimization enabled is configured to send all Fusion Service requests through the Proxy User Data Service.
- Self-Service Optimization enabled user account: When Self-Service Optimization is used, the user is always authenticated by Oracle Identity Cloud Service. A user account with the same User ID value may or may not exist in the Fusion Service Identity Management Store. A user account in Oracle Identity Cloud Service is considered to be Self-Service Optimization enabled only if the GUID of the Oracle Identity Cloud Service user account is stamped on a contact record representing the Oracle Identity Cloud Service user account in Fusion Service.
- Self-Service Optimization user migration: Self-Service Optimization user migration is the process that ensures the GUID of an Oracle Identity Cloud Service user account is stamped on a contact record representing the Oracle Identity Cloud Service user account in Fusion Service.

Pre-Migration Tasks

Overview of Pre-Migration Tasks

Use this topic as a guide as you prepare to migrate your Digital Customer Service implementation to include Self-Service Optimization.

Before you begin, you must previously have set up your environment following the instructions in Chapter 3. Prior to doing user migration, however, you must complete the steps laid out in this topic.

Overview of Tasks

Here's an overview of the tasks you'll perform.

Note: You must perform each of these listed tasks in the order they're presented.

1. Set up Oracle Identity Cloud Service
2. Set up Oracle Fusion Service
3. Set up Administrators and Developers
4. Set up Oracle Visual Builder
5. Set up Oracle Visual Builder Cloud Service with Oracle Fusion Service
6. Prepare your current application for Self-Service Optimization

Pre-Migration Tasks for Setting Up Oracle Identity Cloud Service

First you set up Oracle Identity Cloud Service using the following topics from Chapter 3.

1. Set Up Oracle Identity Cloud Service for Authentication.
2. Create the Application Client.
3. Allow Anonymous Users to Access the Signing Certificate.
4. Configure the Password Reset Email.
    - You use the Resend Welcome template when new users are registered by way of Self Service Optimization. It's recommended that you ensure the message content is appropriate.
    - You use the Password Recovery template when the user has been migrated. This template is also used when user uses the forgot password flow and so, if this template is configured for the purpose of migration, it must be changed appropriately after migration is complete.

      Note: If users are presently using IDCS as the Identity Provider then disable the notification based on the Password Recovery Request template so that users don't receive an email asking them to change their password.
5. In Identity Cloud Service disable the notification called User Profile Updated by Administrator. By disabling this notification, you ensure that email isn't sent about the user profile change when the isFederated parameter is set to FALSE. You do this in the menu under Settings, Notifications and then by deselecting User Profile Updated by Administrator and then saving the setting.
6. Use the Fusion Service application in Oracle Identity Cloud Service to trigger a full import of users and roles from Fusion Service to Oracle Identity Cloud Service. Once the operation is complete, check the Synchronization Failure Report to determine if there were any issues that need to be addressed.

Pre-Migration Tasks to Set Up Oracle Fusion Service

Now you set up Oracle Fusion Service using the following tasks from Chapter 3.

1. Configure a User Category for Proxy Users.
2. Create the Proxy Users.

> **Note:** If you're using custom job roles for your self-service users, consult the Related Topics for a link to the guidelines on creating proxy users.

3. Set Profile Options for Self-Service Optimization.

> **CAUTION:** In the production environment, change the value of profile option SVC_CSS_USE_FA_AS_IDP to false only just before you migrate the users otherwise user registration may fail or users wont be able to use the application.

4. Set the Oracle Identity Cloud Service Endpoint.
5. Configure the Client ID and Client Secret.
6. Manage the Proxy User Configuration Data.
7. Set Proxy User Credentials.

Related Topics

- Create the proxy users

Pre-Migration Tasks to Set Up Administrators and Developers

While you're developing your application for Self Service Optimization, you test the application in Preview mode using a Developer user account. You create this account by reviewing the information the topic shown in Related Topics.

Related Topics

- What are the Digital Customer Service Developer roles?

Specify Fusion Service Details to Visual Builder Cloud Service

You set up Oracle Visual Builder using the following steps from Chapter 3.

To specify Fusion Service details to Oracle Visual Builder, do the following:

1. Sign into Oracle Visual Builder as an administrator.
2. Click the Menu icon, and select Settings to open the **Tenant Settings** page.
3. Click the **Services** tab, then click the **Back ends** icon (+)
4. In the **Back End Service Type** window, select Oracle Cloud Application Instance, and click **Close**.
5. In the Instance URL field of the Create Oracle Cloud Application Instance window, enter the instance URL of your Oracle Cloud Applications back end service.
6. From the **Authentication** drop down list, select **Oracle Cloud Account**.
7. Click **Create**.

> **Note:** You must set the Instance URL field with the fully qualified domain name of your Oracle Applications Cloud Fusion Service instance.

Prepare Your Current Application for Self-Service Optimization

Now make a new version of the current application and modify it to work with Self-Service Optimization by performing the following tasks:

1. Perform the steps in the Chapter 11 section Create an Application for Self-Service Optimization with Pre-20C Template.
2. Perform the steps in the Chapter 4 task Add Mappings to User Roles.
3. If you're working on the stage environment, you can stage the application and test it by signing up and using that user for further testing the application.

Related Topics

- Enable the Implicit Grant in the Digital Customer Service Application
- Create Development and Production Application Profiles
- Set Up Service Connections

User Migration

Self-Service Optimization User Migration Prerequisites

The Fusion Service user account used to initiate the Self-Service Optimization user migration process must have the below roles as part of their user roles:

- Custom Objects Administration
- Customer Relationship Management Application Administrator
- Employee
- Sales Administrator

Overview of the User Migration Process

The complex Self-Service Optimization user migration task is performed by a method on the IdpMigrationManager class.

These methods are available to Application Composer Groovy scripts. The user migration is initiated by scheduling the Schedule Custom Groovy Object Functions ESS Job which is configured to trigger a custom object function written on a Custom Object. You can view the results of the migration process in a custom report.

From the development perspective, here's what the migration entails:

- Determine a logic for partitioning users that will be migrated.
- And the following three tasks are covered under the Create Required Artifacts for the Migration topic.
    - Create a Custom Object

- Create Custom Object Functions
- Create a Custom Report

Once these pieces are in place, you can schedule the Schedule Custom Groovy Object Functions ESS Job. Once this job is run, you can view the custom report.

Partition Users for Self-Service Optimization

It is strongly recommended that you use the functions available for the Self-Service Optimization user migration and migrate no more than 2000 users at one time.

If you need to migrate more than 2000 self service users you must partition them into batches. This task shows you how to partition users based on the contact party ID value. The users are partitioned into batches and the party ID of the first contact and last contact in the batch are shown. This migration strategy assumes that no users have been migrated. If some users have already been migrated then this task will still work, but if a large number of users have already been migrated this partitioning method may not work as expected.

> **Note:** See the section "Create a Data Model from a Custom Query" for more information on how to run the following queries.

First, for information only, use the following query to find the total number of users:

```
SELECT count(*)
FROM fusion.svc_self_service_roles
WHERE relationship_type_cd = 'ORA_CSS_USER'
 AND delete_flag = 'N'
 AND current_idp_cd != 'ORA_CSS_IDP_IDCS'
```

Next, use the following query that uses the DENSE_RANK analytic function to partition the users into batches. The batch size in this example is 2000.

```
SELECT
 batch_num,
 MIN(contact_party_id) first_contact_party_id,
 MAX(contact_party_id) last_contact_party_id,
 COUNT(*) batch_size
FROM
 (
 SELECT
 contact_party_id,
 floor((rank - 1) / 2000) batch_num
 FROM
 (
 SELECT
 DISTINCT contact_party_id,
 DENSE_RANK() OVER(
 ORDER BY
 contact_party_id
 ) AS rank
 FROM fusion.svc_self_service_roles
 WHERE
 relationship_type_cd = 'ORA_CSS_USER'
 AND delete_flag = 'N'
 AND current_idp_cd != 'ORA_CSS_IDP_IDCS'
 ORDER BY contact_party_id
 )
 )
```

```
GROUP BY batch_num
ORDER BY batch_num
```

The output shows the party ID of the first and last contact in each batch.

Create a Custom Object for the Migration

You first create a custom object as part the migration requirements.

1. Sign in to Oracle Fusion Service as a user who has privileges to perform migration tasks.
2. Create a sandbox by doing the following:
    a. Click **Navigator > Configuration > Sandboxes**.
    b. Click **Create Sandbox**.

       The **Create Sandbox** page appears.
    c. Enter a name in the **Name** field.
    d. Click **Create**.
    e. In the **Available Sandboxes** list, click the name of the sandbox name that you specified in step c.
    f. Click the **Plus** icon (Active Tools).

       The **All Tools** dialog appears.
    g. Click the **Application Composer**, and then click **OK**.
    h. Click **Enter Sandbox**.
3. Navigate to the **Application Composer**.
4. From the Objects list in the Object Explorer, click the **Create (+)** icon against Custom Objects to create a custom object.
5. In the Create Custom Objects dialog box, enter: PudsUserMigrator as the display label. The other required fields will auto fill.
6. Click OK.

Overview of the IdpMigrationManager

The IdpMigrationManager class provides methods that enable existing users in Fusion Service Identity Store to use Self-Service Optimization.

The following table lists the available methods:

Instance Method	Purpose
migrate()	Use this method to migrate all the users.
migrate(contactPartyId: String)	Use this method to migrate only the user identified by the given contact party ID.
migrate(contactPartyIds: String[])	Use this method to migrate all the users identified by the given contact party IDs.

Instance Method	Purpose
migrateByAccountPartyIds(accountPartyIds String[])	Use this method to migrate all the users related to the accounts identified by the given account party ID.

The getInstance() Method

You can use the static getInstance() method of the IdpMigrationManager to obtain an instance of the IdpMigrationManager.

This method takes the following argument.

Argument	Description
idpDestination	The value of this parameters must always be ORA_CSS_IDP_IDCS.
batchSize (optional)	This number determines the number of users fetched from the database and processed at one time. The recommended batch size is 100. This parameter value isn't used when methods that accept contactPartyId as parameter.
totalCount (optional)	The total number of users processed. This value should not be greater than 2000.

Groovy Code

Use the following Groovy code to get an instance of the IdpMigrationManager.

```
def idpDestination = "ORA_CSS_IDP_IDCS";
def mgr = oracle.apps.crm.service.css.migrationService.util.IdpMigrationManager.getInstance(idpDestination,
 100, 2000);
```

Here's an example use of the Groovy script code to enable a single contact identified the contact party ID for Self-Service Optimization.

```
def contactPartyId = 1234L;
def idpDestination = "ORA_CSS_IDP_IDCS";
def mgr = oracle.apps.crm.service.css.migrationService.util.IdpMigrationManager.getInstance(idpDestination);
def msg = mgr.migrate(contactPartyId);
return msg;
```

Create Custom Object Functions

Now you create the following functions:

Function Name	Purpose
getContactsInRange	Returns the party IDs of contacts within a given range.
migrateContactsInRange	Migrates all the contacts in the given range.

Function Name	Purpose
migrateBatch <N>	Migrates a batch of users. You must create one function for each batch you run.

1. In Application Composer, expand Custom Objects, expand the **PudsUserMigrator** node.
2. Select **Server Scripts**, then click the **Object Functions** tab.
3. Click the **Action** drop down list, and select **Add a New Object Function**.
4. In the Create Object Function page, do the following:
    a. In the Function Name field, enter: **getContactsInRange**
    b. Click the Returns drop down list, and select: **List**.
    c. Expand the Parameters area, and click the **Add Parameter** icon.
    d. In the Name field, enter: **start** and from the **Type** drop down list, select **Long**.
    e. Click the **Add Parameter** icon again.
    f. In the Name field, enter **end**, and from the **Type** drop down list, select **Long**.
    g. In the Edit Script field, paste the following code:

```
def partyList = [];
def selfRegnVO = newView('SelfRegistrationVO');

selfRegnVO.appendViewCriteria("""
(ContactPartyId between '${start}' and '${end}' ) AND StatusCd = 'ORA_CSS_APPROVED'
""")

selfRegnVO.setMaxFetchSize(6000)
selfRegnVO.executeQuery()

while (selfRegnVO.hasNext()){
 def curRow = selfRegnVO.next();
 partyList.add(curRow.ContactPartyId);
}
partyList = partyList.unique()
return partyList.sort();
```

    h. Click **Save and Close**.
5. Click the **Add a New Object Function** icon, and In the Create Object Function page, do the following:
    a. In the Function Name field, enter: **migrateContactsInRange**
    b. Click the **Returns** drop down list, and select: **String**.
    c. Expand the Parameters area, and click the **Add Parameter** icon.
    d. In the Name field, enter: **start** and from the **Type** drop down list, select **Long**.
    e. Click the **Add Parameter** icon again.
    f. In the **Name** field, enter **end**, and from the **Type** drop down list, select **Long**.
    g. In the **Edit Script** field, paste the following code:

```
def contactPartyIdList = getContactsInRange(new Long(start), new Long(end)) as Long[];


if(contactPartyIdList.size() == 0){
 throw new oracle.jbo.ValidationException('There are no contacts identified by the given partyId
 range: ' + start + "-" + end);
}

if(contactPartyIdList.size() > 2000){
```

```
 throw new oracle.jbo.ValidationException('There are more than 2000 contacts identified by the
 given partyId range: ' + start + "-" + end);
}

def contactPartyIds = contactPartyIdList as Long[];
def idpDestination = "ORA_CSS_IDP_IDCS";
def mgr =
 oracle.apps.crm.service.css.migrationService.util.IdpMigrationManager.getInstance(idpDestination,
 100, 2000);
def msg = mgr.migrate(contactPartyIds);
return msg;
```

h. Click **Save and Close**.

6. Click **Add a New Object Function**, then in the Create Object Function page, do the following:
   a. In the Function Name field, enter: **migrateBatch<N>**.
   b. Click the **Returns** drop down list, and select: **String**.
   c. Change the Visibility value to **Callable by External System**.
   d. In the **Edit Script** field, paste the following code:

```
def batchStartContactPartyId = ?L;
def batchEndContactPartyId = ?L;

def msg = migrateContactsInRange(new Long(batchStartContactPartyId), new
 Long(batchEndContactPartyId));

return msg;
```

> **Note:** ?L must be replaced with the start and end contact party ID values from the Partition Users for Self-Service Optimization topic. If you save without changing this value, the script will throw an error.

   e. Click **Save and Close**.

> **Note:** Create as many migrateBatch jobs as required based on the partitioning logic output.

Create a Data Model from a Custom Query

Use this topic to create a data model from a custom query.

1. Log in to Fusion Service as a user who can use Reports and Analytics.
2. Click **Navigator > Tools > Reports and Analytics**.
3. Click **Browse Catalog**.

   Your browser opens the Oracle Business Intelligence catalog in a new browser tab.
4. Click **New**, and then from the drop down list, select **Data Model**.
5. On the Diagram tab, click the Plus icon, and select SQL Query from the drop down menu, and do the following:
   a. In the **Name** field, enter: **Migration Data**.
   b. From the **Data Source** drop down list, select **ApplicationDB_CRM**.
   c. From the **Type of SQL** drop down list, select **Standard SQL**.
   d. Paste the following SQL statement into the SQL query text box.

```
SELECT
 event_id,
```

```
 job_id,
 contact_party_id,
 hzp.party_unique_name contact_name,
 event_type_cd,
 etlookup.meaning event_type,
 event_status_cd,
 eslookup.meaning event_status,
 event_note,
 login_id,
 history.last_update_date
FROM
 fusion.svc_css_idp_migr_history history,
 fnd_lookups etlookup,
 fnd_lookups eslookup,
 hz_parties hzp
WHERE
 etlookup.lookup_code = event_type_cd
 AND etlookup.lookup_type = 'ORA_SVC_CSS_EVENT_TYPE_CD'
 AND eslookup.lookup_code = event_status_cd
 AND eslookup.lookup_type = 'ORA_SVC_CSS_EVENT_STATUS_CD'
 AND hzp.party_id = history.contact_party_id
 AND history.last_update_date >= :JOB_START_DATE
```

e. Click **OK**.

6. In the Add Parameter dialog box, select the **JOB_STATE_DATE** check box and click **OK**.
7. In the Parameters page, click the Data Type drop down list for **JOB_START_DATE** parameter and select **Date**.
8. Click the **Mandatory** check box.
9. In the **Display Label** field, enter **Job Start Date**.
10. In the Data Model pane, click **Properties > Data Sets > Migration Data**.
11. Click the gear icon for each entry in the G_1 table, select properties each time, and then enter the information shown in the following table:

Name	Display Name	Data Type
EVENT_ID	Event ID	Long
JOB_ID	Job ID	String
CONTACT_PARTY_ID	Party ID	Long
CONTACT_NAME	Name	String
EVENT_TYPE_CD	Event Type Code	String
EVENT_TYPE	Event Type	String
EVENT_STATUS_CD	Event Status Code	String
EVENT_STATUS	Status	String

Name	Display Name	Data Type
EVENT_NOTE	Comments	String
LOGIN_ID	Login ID	String
LAST_UPDATE_DATE	Last Update Date	Date and Time

12. Click **JOB_ID** and ensure that only that row is selected, then click the **Gear** icon and select **Group By**.
13. Click **CONTACT_PARTY_ID** and ensure that only that row is selected, then click the **Gear** icon and select **Group By**.
14. Click the Save As icon, and give the data model a meaningful name such as Migration DM.
15. Click the View Data button.
16. Enter a value for the Job Start Date and click View to view sample data.
17. Save as sample data.

How do I create a custom report based on a data model?

Now you can create a report based on the data model you have created.

1. Log in to Fusion Service as a user who can use Reports and Analytics.
2. Click **Navigator > Tools > Reports and Analytics**.
3. On the Reports and Analytics page, click **Create**, and then select **Report**.
4. Select Use Data Model, then click the search icon and locate the data model you previously created.
5. Click **Next**.
6. On the **Select Layout** page, choose the **Landscape** page option, and the **Table** layout option, and then click **Next**.
7. On the **Create Table** page, deselect **Show Grand Totals Row** and then click **Next**.
8. On the **Save Report** page, click **Customize Report Layout**.
9. Click **Finish**.
10. In the **Save As** dialog box, enter a name such as **Migration Report**, and then click **OK**.
11. Add a layout grid by doing the following:.
    a. In the report editor, click **Drop a Data Item Here**, and press the delete button on your keyboard.
    b. Click the **Insert** tab (if it's not already selected).
    c. Drag the **Layout Grid** from the Components list and drop it below the report title.
    d. In the Insert a Layout Grid dialog box, enter 1 in the Rows box, and leave the Columns box as 2, and then click **OK**.
    e. Click the **Insert** tab again, and from the Components list, drag a **Text Item** and drop it in the first column.
    f. Double-click the text item and enter the following: **Job started on or after**.
    g. From the **Data Source** pane, drag the **JOB_START_DATE** parameter to the second column.
    h. Reduce the width of the **Job started on or after** column.
12. Add a repeating section component by doing the following:
    a. Click the **Insert** tab, and then drag the **Repeating Section** component and drop it below the Layout Grid.

b. In the **Insert a Repeating Section** dialog box, select **JOB ID** from the **Element** drop down list, and then click OK.
c. Click the **Insert** tab, and drag a **Layout Grid** and drop it into the repeating section.
d. Click the **Insert** tab, and drag a **Text Item** and drop it in the first column.
e. Double-click the text item and enter the following: **Job ID**.
f. From the **Data Source** pane, drag the **JOB_ID** parameter to the second column.
g. Re-size the first column.

13. Now add a data table by doing the following:
    a. Click the **Insert** tab, and drag the **Data Table** component and drop it under the row in the repeating component,
    b. While the data table is selected, click the **Show** drop down list on the **Table** tab and select the first item that indicates no (darkened) summary row.
    c. From the Data Source pane, drag and drop the following fields into the data table:
        - Event ID
        - Contact Party ID
        - Login ID
        - Event Type
        - Event Status
        - Comments
14. Click the **Save** icon, and then click the **Done** button to access the report in View mode.
15. In the **Start Date** field, enter a job start date, and then click **Apply** to view the report.

Schedule the Job for Users for Self-Service Optimization

Before you schedule your job, make sure of the following:

- The value of the profile option SVC_CSS_USE_FA_AS_IDP is set to FALSE.
- All the users to be migrated are in the FA Identity Store and Identity Cloud Service.

Note: The Job Schedule process must be repeated for every migrateBatch <N> custom object function that's created based on partitioning logic.

1. Log in into Fusion Service as a user with privileges required to schedule ESS jobs.
2. Click on **Navigator > Tools > Scheduled Processes**.
3. Click the **Schedule New Process** button.
4. In the Schedule New Process dialog box, enter: **Schedule Custom Groovy Object Functions**, then click **OK**.
5. In the Process Details dialog box, add the Object name (such as PudsUserMigrator_c) and Object Function (such as, migrateBatch1) and click **Submit**.
6. After the job completes, review the log files and the text files.
7. If the job ran successfully, the output log file output will resemble the following:

```
JobId 36773: Successfully invoked and executed PudsUserMigrator_c object function migrateBatch1
```

8. The text file output will include one line for each contact that was migrated successfully. If the status is FAILED you can use this report to figure out what caused it. Here's an example of the output:

```
Executing PudsUserMigrator_c object function migrateBatch1 returns: 300100185580882: MIGRATED
```

View the Migration Report

Use this task to view migration reports.

1. Log in to Fusion Service.
2. Click **Navigator > Tools > Reports and Analytics**.
3. Click **Browse Catalog**.
4. Locate the report you created, and then click **Open**.
5. Enter a value in the **Start Date** field, and click **Apply** to view the report.

Post-Migration Tasks

Overview of Post-Migration Tasks

Use this topic as a guide as you perform post-migration tasks on your Digital Customer Service implementation.

1. Stage the updated Digital Customer Service application that includes the Self-Service Optimization modifications and validate it's functioning correctly for existing users who have been migrated.
2. Publish the updated Digital Customer Service application that includes the Self-Service Optimization modifications.
3. Revert changes in the "Password Recovery Request" Template.
4. Disable the Identity Cloud Service User Sync process if this was only being used for Digital Customer Service and not any other Hybrid SaaS service. Go to Oracle Cloud Services, Click on Oracle Applications Cloud Provisioning and toggle the "Enable Provisioning" to disable it.

12 Alternative Implementation Options

Implement Your Digital Customer Service Experience with REST APIs

Overview of the Digital Customer Service Experience with REST APIs

Find information in this chapter on implementing and configuring your Oracle Digital Customer Service experience using REST APIs.

You will get to know the details on how to develop, configure, manage, and administer an Oracle JET-based customer service application. Also, you will learn how to set up and work with the additional features in Oracle Fusion Service. For more information about Oracle Fusion Service, refer to the Related Topics.

A significant part of implementing your Digital Customer Service experience using REST APIs relies on Oracle JavaScript Extension Toolkit (Oracle JET). Oracle JET is a collection of JavaScript libraries and functions used to develop client based applications. Your implementation will leverage Oracle JET libraries and development best practices to provide service lifecycle, knowledge management, and products association with service requests and user management request and approvals. For more information about Oracle JET, refer to the Related Topics.

The topics in this chapter describe how to build an application purely based on JavaScript using Oracle JET, consuming data from Oracle Fusion Service using REST APIs.

Related Topics

- Oracle Help Center
- Welcome to Oracle JET

Get Started with the Digital Customer Service Experience with REST APIs

Implementing the Digital Customer Service experience with REST APIs requires the JavaScript reference implementation that was developed using Oracle JET and Oracle's JavaScript framework and toolkit, including third-party dependent software.

Installation Prerequisites

You must download all of the dependent software for use on the computer where the Digital Customer Service experience is being implemented with REST APIs.

Note: The versions of the prerequisite software listed in the following table were used in development. Later versions might also work with your deployment.

Software or Package Name	Tested Version
Java Development Kit	1.8.0_181
Node.js	7.10.1
NPM	6.4.1
Oracle JET	5.1.0
Oracle Identity Cloud Service	18.4.2

Related Topics

- Oracle JET
- NPM

Add Self-Service Registration

Users require a user name and password to access the customer service application that is developed on Oracle JET.

If the user is not self-registered, the user must provide a first name, last name, and user name on the Self-Registration page in the customer service application developed on Oracle JET.

Once the required information is provided, the self-service registration process will inform the administrator by sending an email notification. The administrator can then sign in to the application and approve or reject the request. The self-service registration process creates an account in Oracle Fusion Service and assigns appropriate roles. For more information about approving and rejecting self-service registration requests, refer to the Manage Registration Requests in the Related Topics.

Related Topics

- Manage Registration Requests

Set Up Oracle Identity Cloud Service Applications

Overview of Oracle Identity Cloud Service Applications

Your Oracle JET application will access Oracle Fusion Service REST API resources.

Within this model, your Oracle JET application performs actions on behalf of an Oracle JET application user. For authentication to work, the Oracle Identity Cloud Service or other identity provider must support OAuth2-based

authentication. Irrespective of the identity provider used, applications should be defined in the identity provider representing the Oracle JET application. The following topics describe how to create the necessary applications in Oracle Identity Cloud Service.

Follow the instructions in these topics in the order in which they appear:

1. Create an Oracle Fusion Service Resource Application.
2. Create an Application for OAuth Implicit Flow.
3. Create an Application for JSON Web Token Assertion.

Create an Oracle Fusion Service Resource Application

This topic describes how to create a confidential resource application for Oracle Fusion Service in Oracle Identity Cloud Service.

For detailed instructions on creating confidential applications in Oracle Identity Cloud Service, refer to the in the Adding a Confidential Application link in the Related Topics.

To create a confidential resource application for Oracle Fusion Service in Oracle Identity Cloud Service:

1. Sign in to your Oracle Identity Cloud Service administration console.
2. Expand the **Navigation Drawer**, and then click **Applications**.
3. Click **Add**.

   The **Add Application** page appears.
4. Click **Confidential Application**.
5. In the **App Details** pane of the **Add Confidential Application** window, specify a name for the application in the **Name** field.
6. Click **Next** to proceed.

   A confirmation message indicates that the application has been added in a deactivated state.
7. You will be prompted to configure authorization information for your application now. Do not configure now, instead, click **Skip for later**.
8. Click **Next**.
9. Click **Configure this application as a resource server now**, and configure these fields as follows:
   - **Access Token Expiration**. Leave at the default value.
   - **Refresh Token Expiration**. Leave at the default value.
   - **Primary Audience**. Specify the Oracle Fusion Service instance that is used to perform resource invocation.
   - **Add (Allowed Scopes)**. Click **Add** next to **Allowed Scopes**:
     i. In the **Add Scope** dialog enter the following for **Scope**:

        /
     ii. Select **Requires consent**.
     iii. Click **Add**.
10. Click **Next**.
11. Skip the Web Tier Policy prompt by clicking **Skip for later**.
12. Click **Next**.
13. Click **Finish**.

Related Topics

- Add a Confidential Application

Create an Application for OAuth Implicit Flow

This topic describes how to create a mobile application for OAuth implicit flow in Oracle Identity Cloud Service.

> **Note:** You must create a confidential resource application for Oracle Fusion Service in Oracle Identity Cloud Service before following the instructions in this topic. For more information, refer to the Create an Oracle Fusion Service Resource Application topic

For detailed instructions on creating mobile applications in Oracle Identity Cloud Service, refer to the Adding a Mobile Application link in the Related Topics.

To create a mobile application for OAuth implicit flow in Oracle Identity Cloud Service:

1. Sign in to your Oracle Identity Cloud Service administration console.
2. Expand the **Navigation Drawer**, and then click **Applications**.
3. Click **Add**.

   The **Add Application** page appears.
4. Click **Mobile Application**.
5. In the **App Details** pane of the **Add Mobile Application** window, specify a name for the application in the **Name** field.
6. Click **Next** to proceed.

   A message confirms that the application has been added in a deactivated state.
7. In the **Authorization** and **Accessing APIs from Other Application** sections of the **Add Mobile Application** page, configure the these fields as follows:
   - **Authorization Code**. Select this check box.
   - **Implicit**. Select this check box.
   - **Allow non-HTTPS URLs**. Select this check box, and specify the redirect URL.

     > **Note:** If the Oracle JET reference implementation is used, this application already has a callback resource that can be used. For example, the redirect URL would be similar to the following: `http://OracleJETApplicationHost:port/?root=callback` . If the Oracle JET application is not deployed or developed yet, then this field can be left empty temporarily. Once the application is up and running this setting should be updated before using the Oracle JET application for accessing Oracle Fusion Service resources.
8. In the **Allowed Scopes** section, click **Add**.
9. Select the Oracle Fusion Service instance.
10. Click **Next**.
11. Click **Finish**.

Related Topics

- Add a Mobile Application

Create an Application for JSON Web Token Assertion

Use this topic to create a confidential application for JSON Web Token Assertion in Oracle Identity Cloud Service.

> **Note:** You must create a confidential resource application for Oracle Fusion Service, and create a mobile application for OAuth implicit flow in Oracle Identity Cloud Service before following the instructions in this topic. For more information, refer to the Create an Oracle Fusion Service Resource Application and Create an Application for OAuth Implicit Flow topics.

For detailed instructions on creating confidential applications in Oracle Identity Cloud Service, refer to the in the Adding a Confidential Application link in the Related Topics.

To create a confidential application for JSON Web Token Assertion in Oracle Identity Cloud Service:

1. Sign in to your Oracle Identity Cloud Service administration console.
2. Expand the **Navigation Drawer**, and then click **Applications**.
3. Click **Add**.

   The **Add Application** page appears.
4. Click **Confidential Application**.
5. In the **App Details** pane of the **Add Confidential Application** window, specify a name for the application in the **Name** field.
6. Click **Next** to proceed.

   A confirmation message indicates that the application has been added in a deactivated state.
7. Click **Configure this application as a client now**, and configure the these fields as follows:
   - **Client Credentials**. Select this option.
   - **JWT Assertion**. Select this option.
   - In the **Client Type** section follow these steps:
     i. Click **Trusted**.
     ii. Click **Import**.
     iii. Specify the public certificate file that you want to import for use with Oracle JET.
   - In the **Allowed Scopes** section, follow these steps: .
     i. click **Add**.
     ii. Select the Oracle Fusion Service instance.
8. Click **Next**.
9. Click **Finish**.

Related Topics

- Add a Confidential Application

Set Up the JSON Web Token Application

Overview of JSON Web Token Applications

OAuth JSON web tokens can be used to exchange Oracle Identity Cloud Service access tokens. This is a standalone JSON web tokens Java application that returns JSON web tokens which can be used by the Oracle JET application to fetch access token from Oracle

- **User Assertion**. In this model a non-null JSON web token is returned for user JSON web token and a null value for client (Oracle JET application) JSON web tokens token.
- **Client Assertion**. In this model , a non-null JSON web token is returned for both user and client JSON web tokens.

Prerequisites for Running JSON Web Token Applications

Before downloading and running the JSON web token application, Maven and Java Development Kit must be installed on the computer on which the application is deployed. The application was developed using the following versions:

- Maven 3.5.4
- Java Development Kit 1.8.0_221

Note: After installing the prerequisite software, they must be added to the path.

Compile and Start the JSON Web Token Application

This topic describes how to compile and start the JSON web token application.

After downloading the JSON web token application software run the following commands on the command line, in the stated order:

1. `mvn clean`
   This command cleans the project.
2. `mvn install`
   This command compiles the code and creates a jar file in the target directory.
3. `java -jar target/JWTAssertionTokens-1.0.jar`
   This command starts the jetty server to run on port 8080 with the application deployed.

Test the JSON Web Token Application

This topic describes how to test your JSON web token application.

Once your JSON web token application is running you should verify that it's deployed properly and is in working order by testing some URLs.

Note: Alternatively, you can verify the URL using Postman, a free third-party software desktop tool, or any other API testing tool of your choice.

Verify the Output for Client Assertion

Type the following URL into a browser:

```
http://hostname:8080/jwttokens?
userId=username@oracle.com&clientId=ee7323169a7743fa929d94a117b131bb&assertionType=clientAssertion
```

Output similar to the following should be returned for the previously mentioned URL:


```
{
 "userJWTToken" :
 "eyJ4NXQjUzI1NiI6InBMQzNBWGE3b1oxVEpJU2JjcVNvX1BxTVM3cDRsV0VBLVBCZGhydVBHZFkiLCJ4NXQiOiIwRDVLV0pSVVl3NUM5UThvV3BmYX
pSvA_nNC0nH7zgWPel2HQEbTKu7N8BqxnfhbsH8BStFgXq8ctJHT90M-
HvOt0MZkLqb0c6HNQkVMgrL16by7Xxd35rHEjVXe6HnAKQY3b3vmtXgxJRpTCYBhTyOMFmjda4E7xuY8Z2m9H3bkXFGTRbqfWbrgH-
_4VNJK4gWuURpm8a8JhPJtXzH3CifUEP26Wd5AIJFlUlOZs8thQEQgkmas-2WSIZCy4rHtnRjSu3H56GFJ6Fh9Ho1vlPFKznByuFYL_58d1IMI7pW8DJ
 "clientJWTToken" :
 "eyJ4NXQjUzI1NiI6InBMQzNBWGE3b1oxVEpJU2JjcVNvX1BxTVM3cDRsV0VBLVBCZGhydVBHZFkiLCJ4NXQiOiIwRDVLV0pSVVl3NUM5UThvV3BmYX
PdqM46ihSi-rHD3ZBJ4jz5p3e1U2AsqS5jeZrqM6SUw5O_Qrgv3DH-
Ii9RGvFynrP9qEiOl5_UrHjLm6IyTS_DV5AEJ9hEXVmN3H1oMnpgmCqLKPJJfKmpLIASM_QYXywucJRB8RhjZy-
nPD8UKWjuPNmctFmgLsjl2jRv8rRtU9g48A-
Fp55QGRlguRziFhR3YGD2FEPiDjkfw5Cq3gVe5chAhsbxytatKiB5xitwCt4BQRFpTUkfLRUwlXMLP2a7RGHUyfLG9ErgJEB3O6064d_sHo1o5KlxOtD
}
```


Verify the Output for User Assertion

Type the following URL into a browser:

```
http://hostname:8080/jwttokens?
userId=username@oracle.com&clientId=ee7323169a7743fa929d94a117b131bb&assertionType=userAssertion
```

Output similar to the following should be returned for the previously mentioned URL:


```
{
 "userJWTToken" :
 "eyJ4NXQjUzI1NiI6InBMQzNBWGE3b1oxVEpJU2JjcVNvX1BxTVM3cDRsV0VBLVBCZGhydVBHZFkiLCJ4NXQiOiIwRDVLV0pSVVl3NUM5UThvV3BmYX
F_KMnCZcsSH2uhu7rc0_i5-NYnzV_wKgjjIWazVOxSZoU_QJq-4mhCY2cNMkA8YYScNIFzggCFd9ZSrAf1z18Uy6cFeR9-
Z04p2rsCU89SXkAOnN4lHDDD3njWOWmJLFImG8udms2aNA8e2wUFdS_VI3QXkGhFYAz19BDq16lKKyqOmokocHdJapQdy1tTvNGorBDH5KdvkPRYQ_bS
 "clientJWTToken" : null
}
```


Referenced Third-Party Libraries

This topic lists the reference third-party libraries.

The following table lists the referenced third-party libraries for JSON web tokens.

JAR File Name	Description
`com.nimbusds:nimbus-jose-jwt:jar` version 6.7	Java library for Javascript Object Signing and Encryption (JOSE) and JSON Web Tokens (`JWT`).
`commons-codec:commons-codec:jar` version 1.11 `org.apache.commons:commons-lang3:jar` version 3.9	Simple encoder and decoders for various formats such as `Base64` and `Hexadecimal`. Apache Commons Lang, a package of Java utility classes for the classes that are in the `java.lang` hierarchy.

JAR File Name	Description
`org.junit.jupiter:junit-jupiter-engine.jar` version 5.4.2	For unit tests.
`com.fasterxml.jackson.core:jacksc databind.jar` version 2.9.9	Jackson is a high-performance JSON processor (parser, generator).

`org.eclipse.jetty:apache-jsp:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-annotations:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-continuation:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-http:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-io:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-jndi:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-plus:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-security:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-server:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-servlet:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-webapp:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-xml:jar:9.4.15.v20190215`
`org.eclipse.jetty:jetty-util:jar:9.4.15.v20190215`
`org.eclipse.jetty.toolchain:jetty schemas:jar:3.1`
 	
All versions 9.4.15.v20190215
	Jetty-embedded server to serve REST endpoint.

`org.glassfish.jersey.core - jersey-server`
`org.glassfish.jersey.inject - jersey-hk2`
`org.glassfish.jersey.containers - jersey-container-servlet-core`
`org.glassfish.jersey.containers - jersey-container-jetty-http`
`org.glassfish.jersey.media - jersey-media-moxy`
 	
All versions 2.28
	Open source framework for developing RESTful Web Services in Java.

Run the Oracle JET Application

This topic describes how to run the Oracle JET application.

Once configured, your Oracle JET application it's ready for use. We can now build and deploy the Oracle JET application.

To build and deploy the Oracle JET application, run these command in the stated order:

1. `npm install`

   This command installs all `npm` packages that are necessary for running the Oracle JET application. Run this command whenever there are changes to the packages used by the application
2. `ojet build`

   This command builds the Oracle JET application
3. `ojet serve`

   This command deploys the Oracle JET application on port 8000.

   > **Note:** You can deploy your Oracle JET application to a different port by using the command with the following parameter: **--server-port=<port#>**. Where <port#> is the port number.

Configure the Oracle JET Application

Use this topic to configure your Oracle JET application.

After downloading the Oracle JET application, you must configure the javascript file to include site-specific values. The javascript file can be found in the installed directory, at the following path:

`../src/js/config/siteConfig.js`

The configuration file is seeded with default values that represent the authentication scheme chosen to connect to Oracle Fusion Service resources, application endpoints, and timeout values. The following table provides an overview of the configuration parameters that require modifications.

Property Name	Description
`adminUserForRegn`	The user name used by self-signed user registration in conjunction with `adminUserPswdForRegn`, the password to sign in to Oracle Fusion Service. For example: `firstname.lastname@yourdomain.com` **Note:** This property is used exclusively for self-service registration.

Property Name	Description
`adminUserPswdForRegn`	The password used by self-signed user registration in conjunction with `adminUserForRegn`, for the user name to sign in to Oracle Fusion Service. **Note:** This property is used exclusively for self-service registration.
`faInstance`	The URL for Oracle Fusion Service.
`loginType`	A string that represents how your Oracle JET application connects to Oracle Fusion Service. These are possible values: • **Basic** When the user attempts to login, the user name and password are required. • **Implicit** Uses the OAuth Implicit grant type flow. This method is browser-based, where a user is redirected to the identity provider to sign in. With this method, the user consents to allowing Oracle Fusion Service to access resources on their behalf. • **UserAssertion** Uses JSON web tokens provided by a jetty-based server application. The JSON web tokens are used as user information. The client ID and client secret are used to represent the application. • **ClientAssertion** - Uses JSON web tokens provided by a jetty-based server application. JSON web tokens are used for the user and client.
`authzEndpoint`	The URL for the Oracle Identity Cloud Service authorize endpoint. For example: `https://IDCSservice/oauth2/v1/authorize` **Note:** This property is used only when `loginType` is set to **Implicit**.
`idcsTokenEndpoint`	The URL for the Oracle Identity Cloud Service token endpoint. For example: `https://IDCSservice/oauth2/v1/token/`
`redirectUri`	The URL for the Oracle JET application callback endpoint. For example: `http://OJETApplication:portnumber/?root=callback`
`clientId`	The string representing the Oracle Identity Cloud Service application ID. For example: `8dab63f5fc204a3582c80fc76ccba1e3`
`scope`	The URL representing the Oracle Fusion Service instance.

Property Name	Description
`assertionClientId`	The string representing the Oracle Identity Cloud Service application ID of a confidential application. For example: `he7323169a7743fa929d94a117b138bb`
`clientSecret`	The string representing the Oracle Identity Cloud Service application client secret. For example: `64f25f01-b7d7-4a50-9b30-9ea9c8967c8b`
`jwtToken`	The URL for the Oracle JET application endpoint. For example: `http://JWTTokensAppHost:portnumber/jwttokens` **Note:** This property can be left blank when `loginType` is set to **Implicit** or **Basic**.
`serviceRequestResource`	The path for service request resources. For example: `/crmRestApi/resources/11.13.17.11/serviceRequests/`
`useApiGatewayService`	If autonomous API Gateway service is used then set this property to **True**, if not, set it to **False**. Possible values are **True** or **False**.
`apiGatewayEndpoint`	The URL endpoint for the autonomous API Gateway service. For example: `http://APIGatewayServiceHost:8001`

Implement Web Chat

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)

Define a Web Chat Component

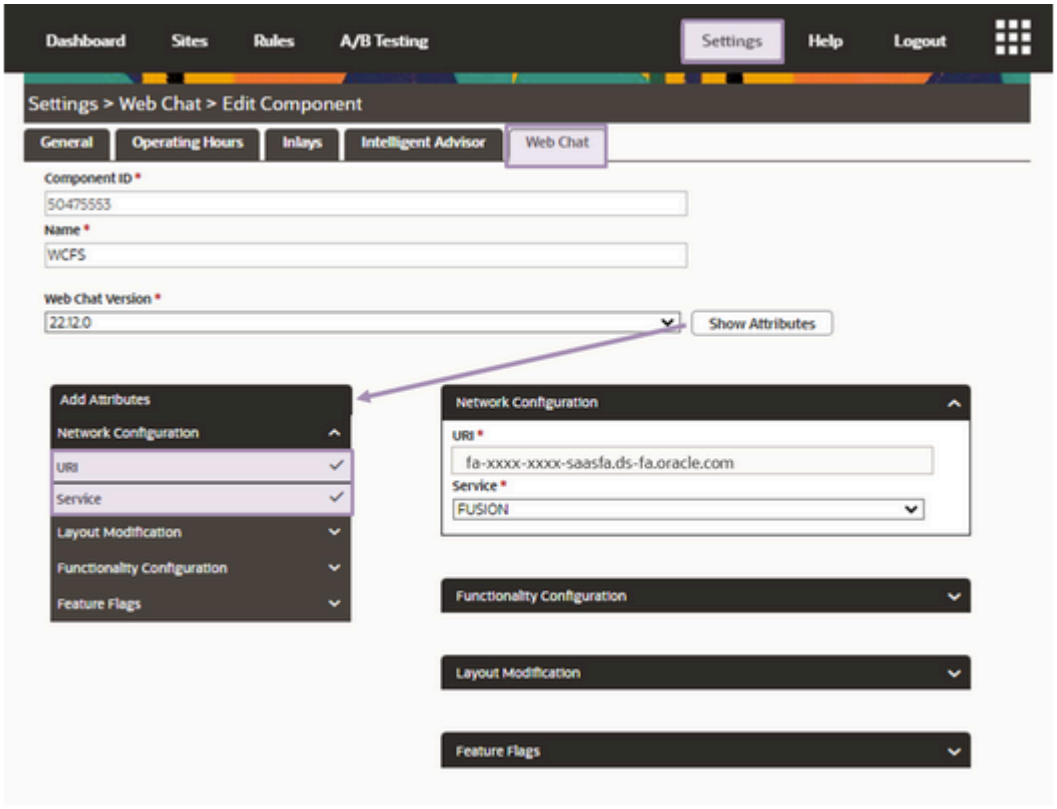
You can configure your web chat widget using the Engagement Engine user interface to be consistent with the branding of your site and your service process.

Here's the different configuration tasks you can perform on the component attributes:

- Network Configuration
- Layout Modification
- Functionality Configuration
- Feature Flags

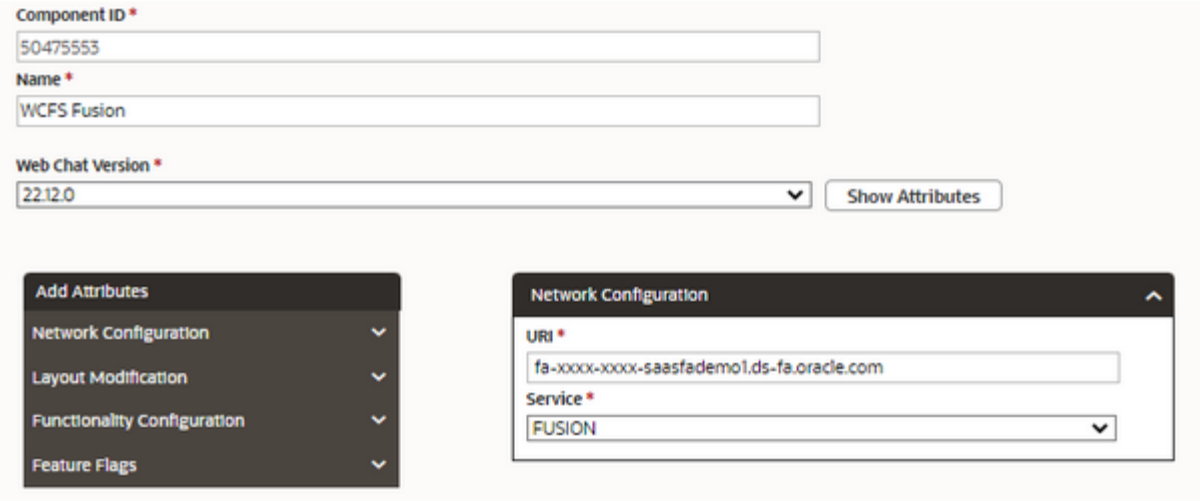
Network Configuration

1. Sign in to the Rule Editor, then go to **Settings > Web Chat > Create New Component**.



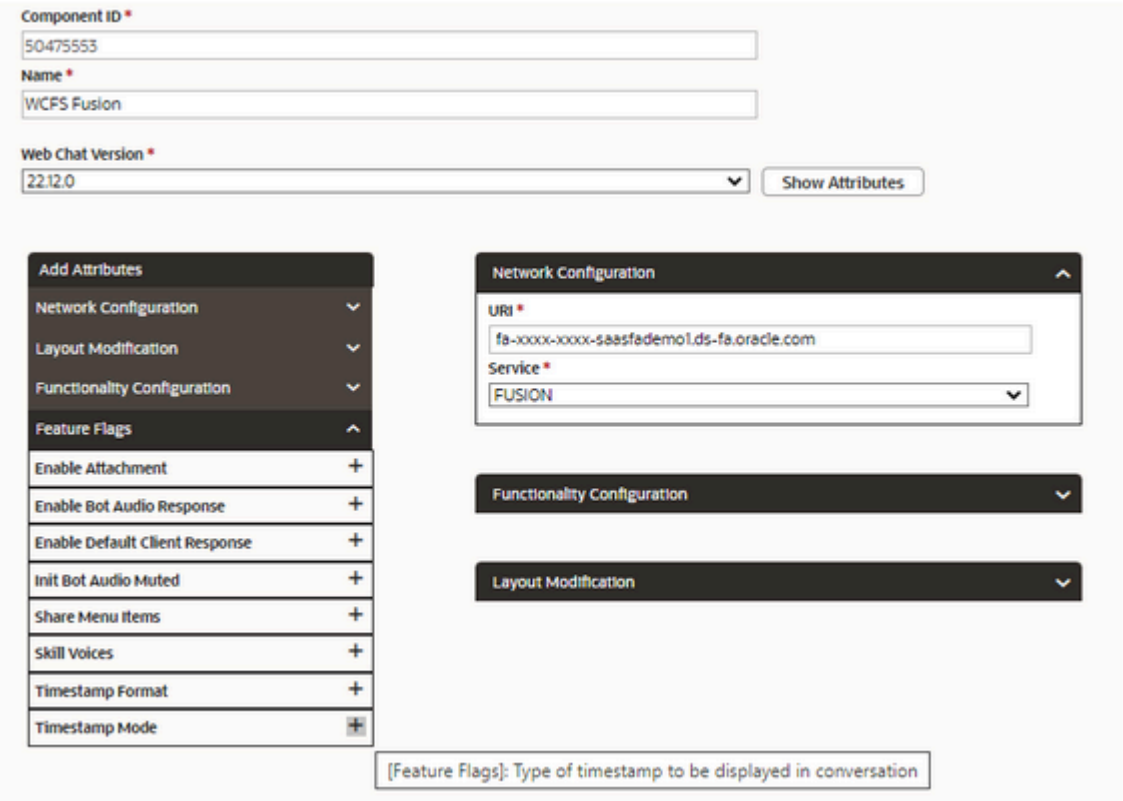
2. Enter a name for the new component, and choose a version.
3. Click **Show Attributes** to view the component attributes for the version you selected.

4. The categorized attribute list appears along with the required attributes for the component.



You can browse through each attribute list by clicking the drop-down list arrow.

You can set the attribute value by clicking the (**+**) icon.

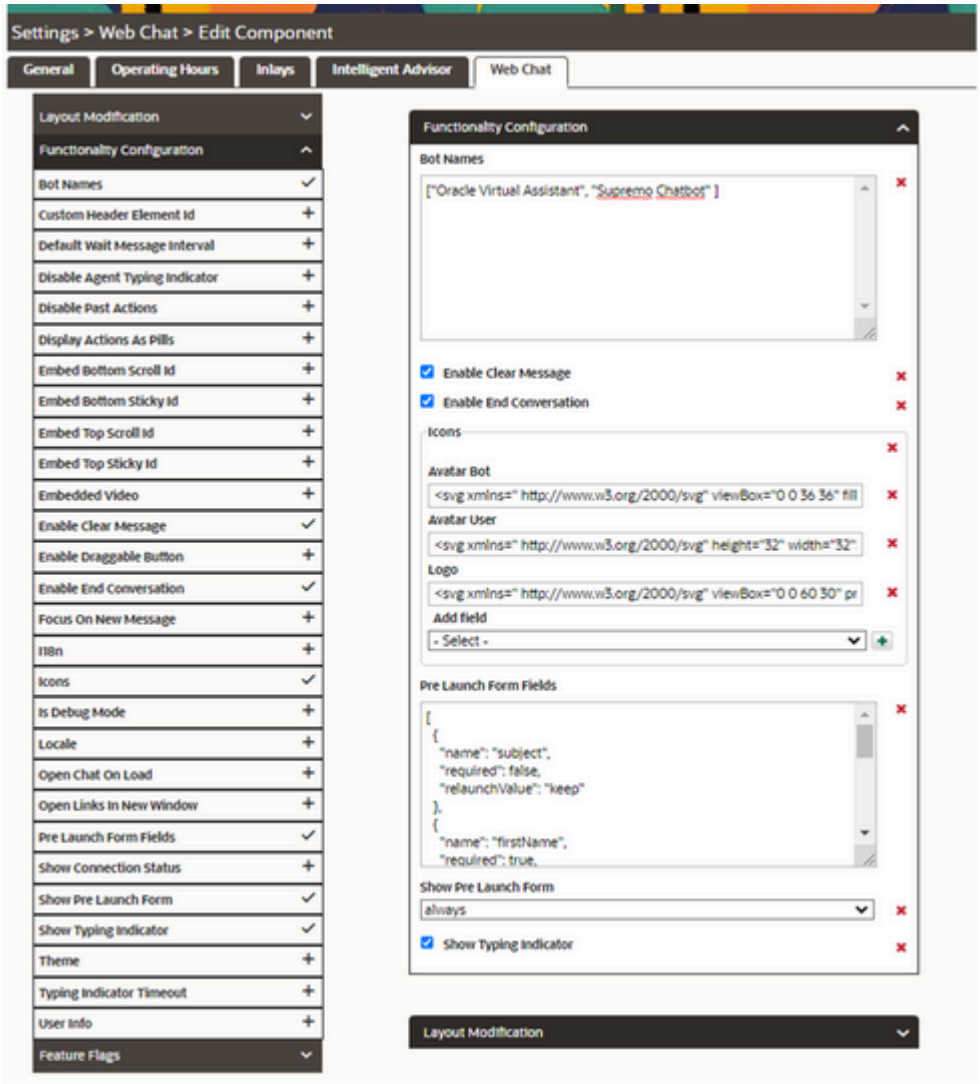


5. In the Network Configuration attributes section, set the required attribute URI with your Fusion instance.

   This value is the full Fusion Service host name without modification (FQDN).
6. Click the Service drop-down list and select Fusion.

Functionality Configuration

1. In the explorer, click the **Functionality Configuration** drop-down list to view attributes for the pre launch form and the icons.



Here's an overview of some attributes:

- **Icons**:
  - Define an image URL for your desired icons to show after starting the conversation
  - The icons used in the widget can be configured in two ways - the URL of the image asset, or an SVG string.

    For the icons that support SVG strings, you can pass the raw SVG data that the widget renders as an inline SVG.

    SVG strings allow for the faster loading of an image, the ability to animate it, or for changing its color.
- **preLaunchFormFields**:

Configure standard attributes to show as fields in the pre launch form

- **showPreLaunchForm**:
  - 'always' - This is the default, every time a new conversation is started the form will be shown before connecting and starting the conversation.
  - 'initial' - The form will only open once, when connecting for the first time. The pre launch form allows the user to interact before starting the conversation and provide supported profile information like an email, a product, and so on.
  - 'never' - Connect and start the conversation without any user interaction.
- Set the fields for the chat launch form via Pre Launch Form Fields attribute.

Any required user information can be entered via a pre launch form which can pre fill fields with default values. Click the icon in the top-right corner to display the default JSON and keep the default fields you want to display on the pre launch form.

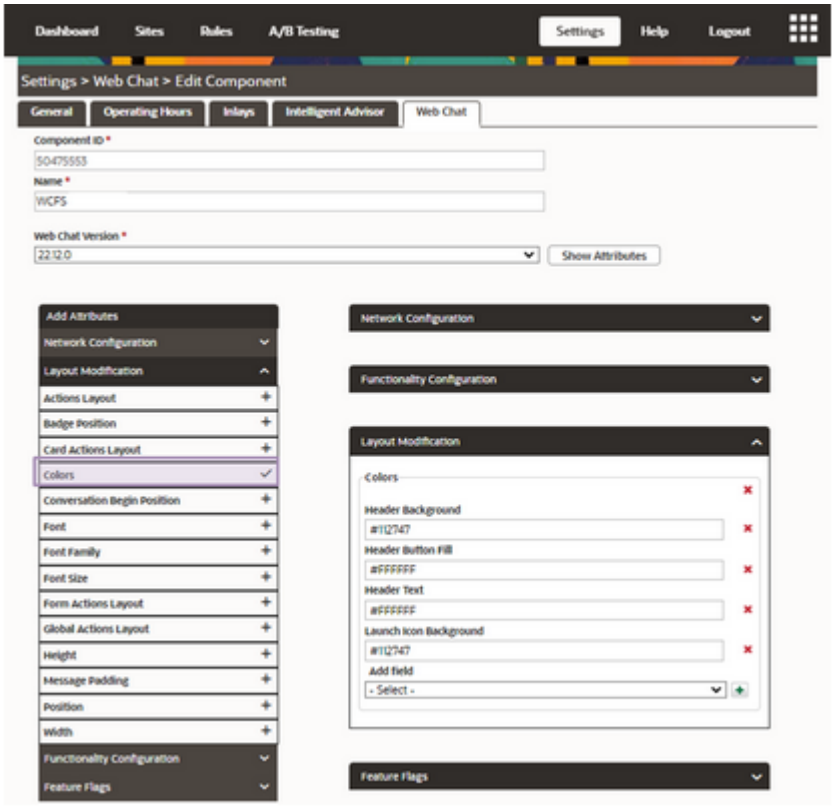
Here's an example:

```
[
 {
 "name": "subject",
 "required": false,
 "relaunchValue": "keep"
 },
 {
 "name": "firstName",
 "required": true,
 "relaunchValue": "default"
 },
 {
 "name": "lastName",
 "required": true,
 "relaunchValue": "default"
 },
 {
 "name": "email",
 "required": true,
 "relaunchValue": "clear"
 },
 {
 "name": "productId",
 "required": false,
 "relaunchValue": "clear"
 },
 {
 "name": "categoryId",
 "required": false,
 "relaunchValue": "clear"
 },
 {
 "name": "queueId",
 "required": false,
 "relaunchValue": "clear"
 }
```

]

Layout Modification:

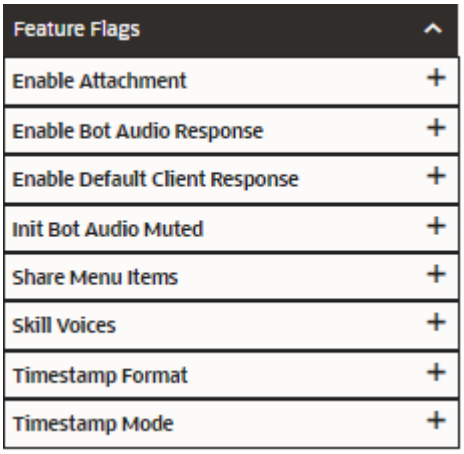
In the Layout Modification attributes section you can define the branding and position of your widget with attributes such as colors, fonts, position width and so on.



Feature Flags:

In the Feature Flags attributes section you can define more settings that are available in Oracle Digital Assistant Native Client SDK for Web such as enabling the user to share location, bot audio response or defining the formatting of the time stamp shown with messages.

> **Note:** Only a subset of the properties available in ODA Native Client SDK for Web are supported in the Web Chat Client.

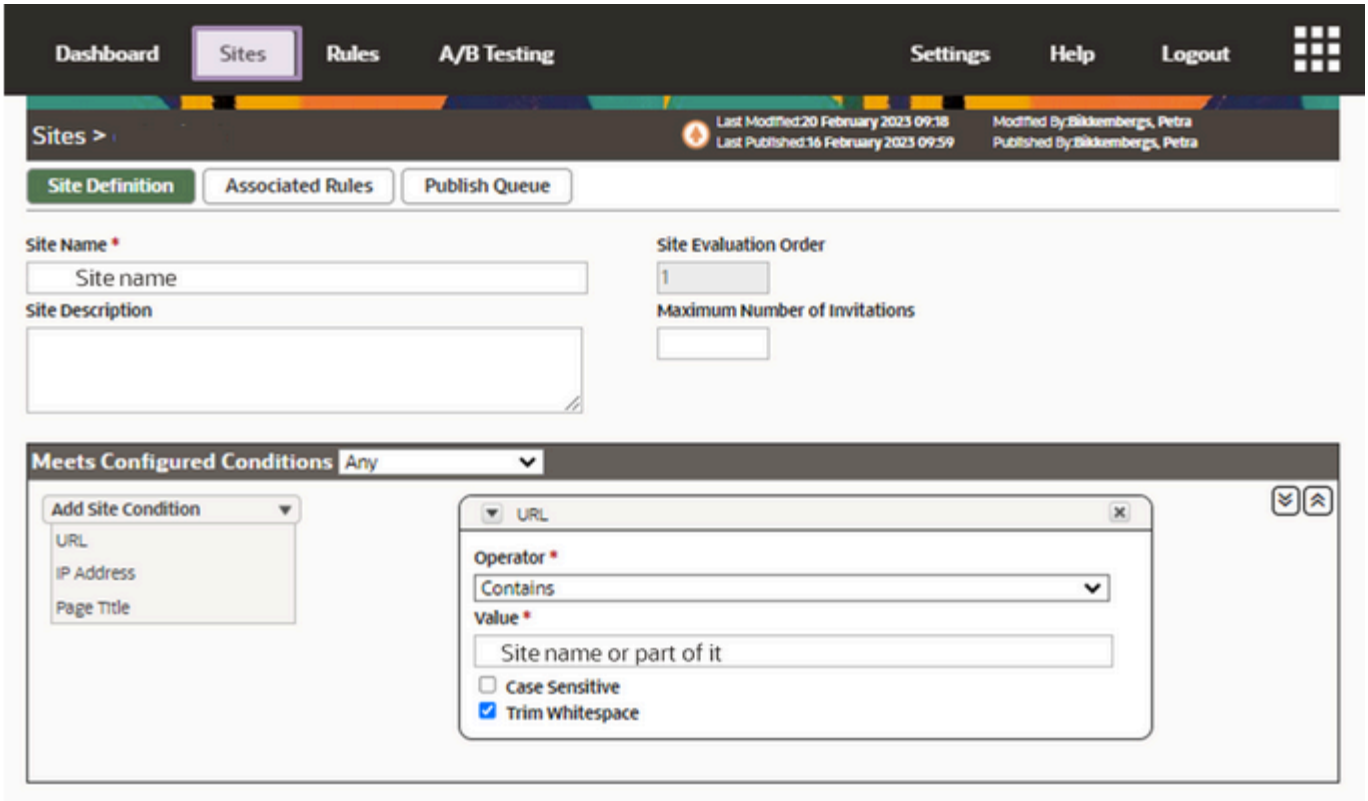


Define Your Site

Now you define the sites where you want to enable the Web Chat Client.

1. Click **Sites** in the navigation bar of Engagement Engine.
2. Click **Create New Site**..
3. Enter a site name in the **Site Name** field.
4. Enter a site description in the **Site Description** field.
5. Click the **Add Site Condition** drop-down list in the **Meets Configured Conditions** section, and then select a condition such as the URL that contains your site name.

6. Save the site.



For more information on Engagement Engine, see:

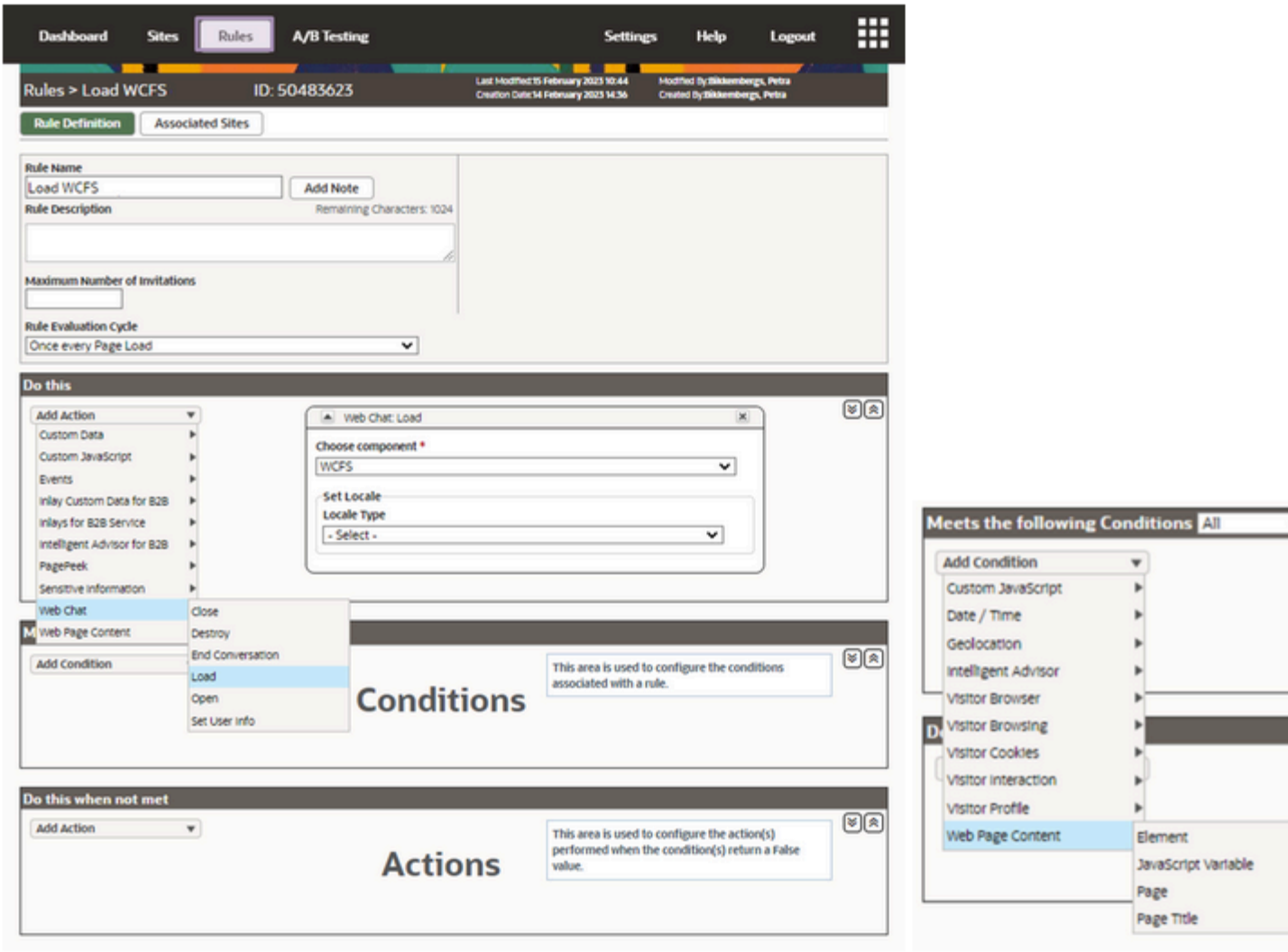
- *Overview of Rules*
- *Site and Rule Association*

Create and Publish a Rule for Web Chat

Now create a new rule to load Web Chat.

1. Click **Rules** in the navigation bar of Engagement Engine
2. Click **Create New Rules**.
3. Enter a rule name in the **Rule Name** field.
4. Enter a description in the **Rule Description** field.
5. Click the **Rule Evaluation Cycle** drop-down list and select the required evaluation cycle.
6. Click **Add Action** in the **Do this** area, and then select **Web Chat > Load**.
7. Select one of the Web Chat components from the **Choose component** drop-down list.
8. Click **Add Condition** in the Conditions area and add a condition (optional)
9. Select the actions you'd like to perform in the **Do this when not met** section if the conditions you define aren't met (optional).

10. Save the rule, associate it with a site, and then publish it.



For more information on Engagement Engine, see:

- *Overview of Rules*
- *Site and Rule Association*
- *Site Publishing*

Engagement Engine Settings

A single page application (SPA) is an application where the page loads only once on the visitor's web browser.

Single Page Application

An SPA is fully loaded on the initial page load and then page regions are updated with new page fragments loaded from the server on demand. Since the page load happens only once with an SPA, a different approach is required when using

the Rules Editor with an SPA. For an SPA, the initial site evaluation and loading of rules is performed when the page loads. Subsequent site evaluations and loading of rules are performed when the hash fragment changes, or when you use the HTML5 pushState method. **See Set the Rules Editor to Work with a Single Page Application.**

The checkbox is located in the Engagement Engine rules editor on the **Settings** tab. You must select this setting for your Digital Customer Service application.

HTTP Referrer URL

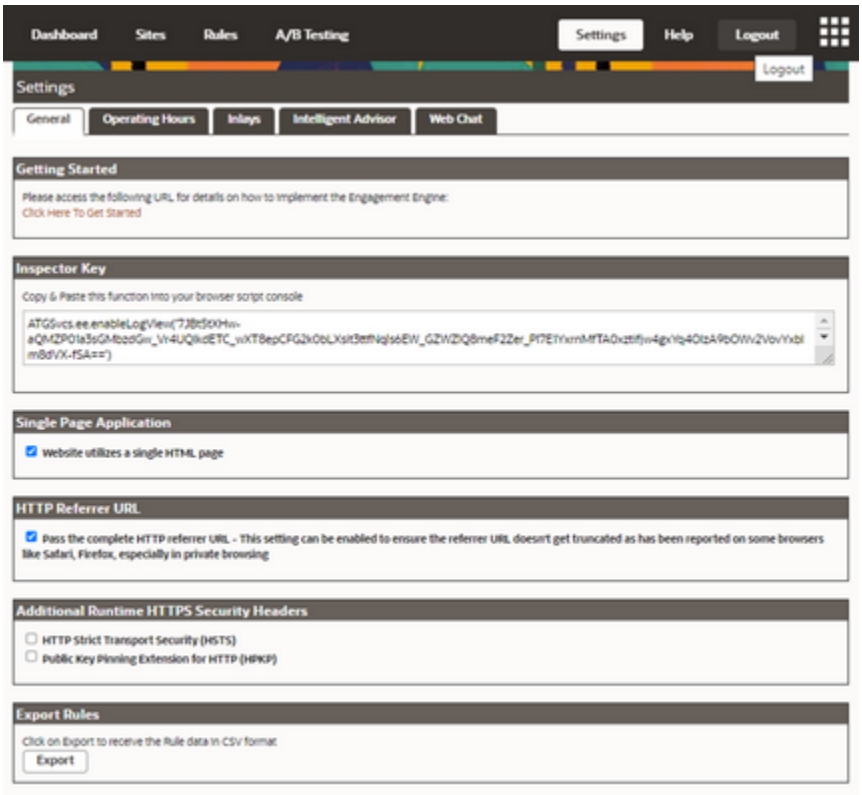
Select this setting to ensure that the complete URL is used for rule and site evaluations.

On some browsers such as Safari, Firefox, and Chrome, especially in private browsing, the HTTP Referrer URL of the page is truncated to display only the domain. This can lead to URL based rule or site conditions to be erroneously evaluated.

For instance, if the page was at https://www.example.com/home/homepage the referrer URL in the request sent by the browser would only be https://www.example.com/ and thus, if the site/rule condition was to check if the URL contained the string "/home/homepage", that condition would wrongly be evaluated to false. This behavior may also be reported for other browsers in any other conditions too, as it is dependent on the browser and its versions.

If this HTTP referrer URL setting is enabled, the referrer URL is ignored and the complete page URL is used by Engagement Engine product for rule evaluation and thus leads to the correct expected results, for both rule and site condition evaluations. The reason for introducing this checkbox was for backward compatibility as we did not want any sudden changes for existing customer's rule conditions.

The checkbox is located in the Engagement Engine rules editor under the **Settings** tab.



For more information on Engagement Engine Rules Editor settings, refer to: *Overview of Settings*.

Add Web Chat for Service to the Digital Customer Service Shell

Web Chat for Service is only usable by organizations that have tagged their pages with the Oracle Engagement Engine JavaScript tag.

We recommend you add the following JavaScript tag in a universal header or footer of your company's test website, so Engagement Engine is available across all pages on the test website.

Replace customer_id with the Engagement Engine account number that Oracle has provided you.

```
<script type='text/javascript'>
window.EESvcs = { eeid:<customer_id };
</script>
<script type='text/javascript' src="//ee.channels.ocs.oraclecloud.com/js/eesvcs.js">
</script>
```

To deploy the Web Chat for Service widget on your Digital Customer Service application, you must add the Engagement Engine tag on your shell page.

You can add the full tag code as described previously in this topic, but for DCS we recommend using the standard Engagement Engine component available from the Component Exchange.

Here's the steps to deploy Engagement Engine on DCS, Visual Builder app:

1. Create a new *Digital Customer Service Application*.
2. In your application, Select the **shell** page under the **Root Pages** folder.

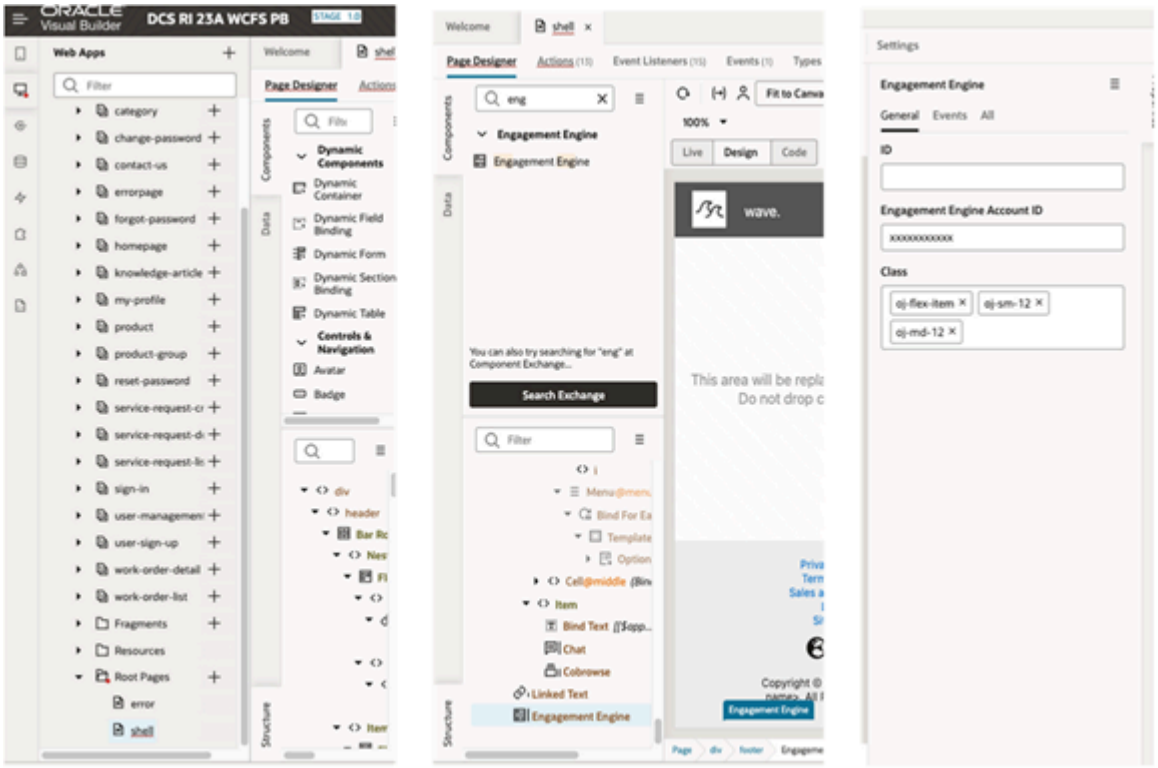
   Adding the Engagement Engine Tag to this page ensures that it's made available in all other pages which inherit the shell page.
3. Now, search for Engagement Engine in the component browser.

   > **Note:** You may be required to install this component from the Component Exchange.

4. Drag and drop the Engagement Engine component from the browser to the bottom of the Shell page in the Design View.

5. With the Engagement Engine component selected on the shell page, add your Engagement Engine Account ID in the Properties editor.

> **Note:** Your account ID is the numeric value assigned to the **eeid** variable located in the script tag.



For more information on Engagement Engine, refer to *Administering Oracle Engagement Engine*.

For information on including page tags on your web page, refer to *Add Page Tags to Your Text Website*.

If you're adding the Engagement Engine component and page tags using Oracle Visual Builder, take a look at *Add the Engagement Engine Component with Visual Builder*.

Enable Profile Options for Web Chat for Service

Web Chat for Service is a syndicated widget that you can deploy on your Digital Customer Service application or any web page through the Engagement Engine single tag.

Web Chat for Service makes use of a chat session that occurs between an end user and a chat agent.

To use Web Chat you must enable a number Profile Options for Chat Fusion Service.

Set Profile Options for Self-Service Optimization

First you must ensure the the correct profile options for Self-Service Optimization are set. Refer to *Set Profile Options for Self Service Optimization* for more information.

Configure Chat Profile Options

Chat profile options enable you to configure assignment and routing options for chat requests. You must configure these chat profile options to enable various chat features.

Enable profile options for Chat to enable the web chat for service widget, allow anonymous access and enable the wait time setting.

Profile Option	Description	Default Value
SVC_CHAT_INLAYS_ACCESS_ENABLED	Enables access to the web chat widget to get the bootstrap configurations. This was a profile option also available for chat inlay.	No
SVC_CHAT_ANONYMOUS_ACCESS_ENABLED	Enables anonymous access to chat for your customers.	No
SVC_CHAT_WAIT_TIME_ENABLED	Enables displaying of the estimated wait time for a customer before an agent is expected to accept the chat request.	Yes

1. In the Setup and Maintenance work area, click the Tasks icon, and then click the **Search** link.
2. In the Search field, enter **Manage Administrator Profile Values**.
3. Click the link for the task.
4. To enable chat interactions do the following:
    a. In the Manage Administrator Profile Values work area, enter the following in the Profile Option Code field: SVC_CHAT_INLAYS_ACCESS_ENABLED and click Search.
    b. Set the Profile Value field to **Yes**.
    c. Click **Save**.
5. To enable anonymous access do the following:
    a. In the Profile Option Code field, enter: SVC_CHAT_ANONYMOUS_ACCESS_ENABLED, and click Search.
    b. Set the Profile Value field to **Yes**.
    c. Click **Save**.
6. To enable the wait time setting do the following:
    a. In the Profile Option Code field, enter: SVC_CHAT_WAIT_TIME_ENABLED, and click Search.
    b. Set the Profile Value field to **Yes**.
    c. Click **Save**.
7. Click **Save and Close**.

Create an Application for Self-Service Optimization with Pre-20C Template

Enable the Implicit Grant in the Digital Customer Service Application

If you're using a template prior to 20C, you may need to perform this step. Enable the implicit grant in the Digital Customer Service applications security settings by doing the following:

1. In Visual Builder, open the Digital Customer Service application.
2. From the navigator panel, select the **Web Applications** icon.
3. Select **dcs** from the list.
4. Click the **dcs** tab in the work area.
5. Click the Security sub tab, and make sure **Enable implicit grant** is selected.

Create Development and Production Application Profiles

Use this task to create Development and Production application profiles.

1. In Visual Builder, open your Digital Customer Service application.
2. Click the **Settings** tab, then click the **Application Profiles** sub tab.
3. For the Default application, click **Rename**.
4. In the Name field, enter **Development**.
5. In the Description field, enter **Default development profile**.
6. Click **Save Changes**.
7. Click the **Duplicate** button, and enter the following values:
    - Name: Production
    - ID: Production
    - Description: Default production profile
8. Click **Duplicate**.
9. Click the menu beside either the Development profile or the Production profile and choose the following options:
    - For Development, choose **Make development default**.
    - For Production, choose both **Make stage default**, and **Make publish default**.
10. Map the user roles for the new Production profile.
11. Complete the Add Mappings to User Roles from chapter 4, adding the Production profile role.

Set Up Service Connections

Now you set up REST API connections.

Configure crmRestApi

First you configure the crmRestApi.

1. Navigate to your Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click Service Connections, click crmRestApi, and then click the Servers tab.
4. Click the **Copy** icon to make a copy the production server. This server has the following values.

Field	Enter this information
Instance URL	vb-catalog://backends/fa/proxy/crmRestApi/resources/11.13.18.05
Description	Production Server.
Application Profiles	Production.
Add custom header	VB-Alt-Authorization-Header-Name with value Puds-Access-Token.
Authentication for logged-in users	Propagate Current User Identity.
Authentication for anonymous users	None.
Connection type	Dynamic, the service supports CORS.

5. For the original server (the server you just copied) click the **Edit** icon and enter the required information listed in the following table:

Field	Enter this information
Instance URL	Keep the URL value.
Description	Development Server
Application Profiles	Development

Field	Enter this information
Security	Select the **Allow anonymous access to the Service Connection Infrastructure** check box.
Authentication for logged-in users	a. Select **Basic** from the drop down list. b. Click the **Enter User name and password** icon in the User name field, and then enter the developer APPID and password, and then click **Save**. **Note:** See Related Topics. In the Set Up Administrators and Developers topic, you will find instructions on setting up the Developer APPID.
Authentication for anonymous users	Click the drop down list, and select **Same as Authenticated User**.
Connection type	Select **Always use proxy irrespective of CORS support**.

6. Click **Save**.

Configure the fscmRestApi (Optional)

If your Digital Customer Service application has an fscmRestApi then perform the following steps.

1. In Oracle Visual Builder, open your Digital Customer Service application (if it's not already open).
2. Click Service Connections, click **fscmRestApi**, and then click the **Servers** tab.
3. Click the **Copy** icon to make a copy the production server. This server has the following values.

Field	Enter this information
Instance URL	The URL value of your instance.
Description	Production Server.
Application Profiles	Production.
Add custom header	VB-Alt-Authorization-Header-Name with value Puds-Access-Token.
Authentication for logged-in users	Propagate Current User Identity.
Authentication for anonymous users	None.
Connection type	Dynamic, the service supports CORS.

4. For the original server (the server you just copied) click the **Edit** icon and enter the required information listed in the following table:

Field	Enter this information
Instance URL	Keep the URL value.
Description	Development Server.
Authentication for logged-in users	a. Select **Basic** from the drop down list. b. Click the **Enter User name and password** icon in the User name field, and then enter the developer APPID and password, and then click **Save**. **Note:** See Related Topics. In the Set Up Administrators and Developers topic, you will find instructions on setting up the Developer APPID.
Authentication for anonymous users	Select **Same as Authenticated User**.
Connection type	Select **Always use proxy irrespective of CORS support**.

5. Click **Save**.

Create an Application with the Standard Digital Customer Service Configuration with a 20C Template or Later

Create an Application without Self Service Optimization using a 20C Template or Later

To create a standard Digital Customer Service application that's not enabled with Self-Service Optimization, using the 20C template, use these tasks.

Set the Default Application Profile

First, you make the Default application profile the default for development, stage and publish.

1. Sign in to Oracle Visual Builder as an administrator.
2. Click the Menu icon, and select Settings.
3. From the Settings page, click the Application Profiles tab.

4. In the Default entry, expand the menu and do the following:
    a. Select Make development default.
    b. Select Make stage default.
    c. Select Make publish default.

Map the Application User Roles

Now you map the application user roles for the Default application profile.

1. In Visual Builder, ensure the Default application is opened.
2. Click the Menu icon, and select Settings.
3. Click the User Roles tab.
4. Click the Application Profile drop down list, and select Default.
5. Refer to the Add Mappings to User Roles topic in the Mandatory Setup Tasks chapter.
6. Add the Customer Self-Service User mapping:
    a. On the **User** card, click **Assign groups or users**.

        The **Change Assignments** dialog box is displayed.
    b. In the list of available mappings, search for the following string:

        Customer Self-Service User
    c. Click the plus sign (**+**) next to **Customer Self-Service User**.
    d. Click **Save Changes**.
7. Add the Customer Self-Service Account Administrator mapping:
    a. On the **Account_Administrator** card, click **Assign groups or users**.

        The **Change Assignments** dialog box is displayed.
    b. In the list of available mappings, search for the following string:

        Customer Self-Service Account Administrator
    c. Click the plus sign (**+**) next to **Customer Self-Service Account Administrator**.
    d. Click **Save Changes**.

Configure Service Authentication

And the final step is to configure service authentication.

1. In the Visual Builder Settings workspace, click the **Services** tab.
2. From the Backends list, select **Oracle Cloud Applications**.
3. Select **Default Oracle Cloud Applications Server without user proxy**.
4. Click the Enter user name and password icon by the User name field.
5. In the User name field, enter the user name of the APPID user.
6. In the Password field, enter the Fusion Service password for the APPID user.

13 Performance

Performance Tuning Checklist

Here's a collection of performance tuning tips. They're listed in no particular order of importance but can be considered once your Digital Customer Service application is up and running.

Potential Issue	Performance Tuning Tip
The application is slow because it hasn't been bundled and minified for deployment.	See the Related Topics area for a link to the Optimize Your Application for Deployment topic.
Large images take too much time to load and are then scaled down for the section of screen within which they're displayed.	Reduce the size of images to the size needed for the UI presentation. Optimize the image for the usage in the UI presentation either by size or quality, balancing small file size against acceptable quality.
Application load times are slow when the client region is far from the data center location of the deployment.	Deploy the application to global Content Delivery Network (CDN). Visual Builder's CDN functionality sets the base **`href`** of the application to the CDN server to ensure all resources are correctly loaded from the CDN. As a result relative URLs including anchor references will be resolved relative to the base **`href`** value. For application navigation it's recommended you use the Visual Builder navigate action and the DCS navigate event. Where **`href="#"`** has been used as a placeholder it should be replaced with **`href="javascript:;"`**. To navigate to an anchor within a page you'll need to use **`<a href="javascript:;" onclick="document.location.hash='myanchor';">`** See the Related Topics section for a link to the topic Host an Application on a Content Delivery Network (CDN).
Page navigation is slow because of URL links and results which are reloading the entire application with each navigation.	Do page navigation with navigation events. For information on using your Navigation Action to help in tuning navigation between pages in a flow, and to root pages, see the Related Topics section for the Navigate Action topic in the Developer Applications with Oracle Visual Builder guide. For an example of how this works, see these action definitions of the service-request-list-start page: **`NavigateToServiceRequestCreateAction`** and**`ListNavigateToServiceRequestDetailClickAction`**
REST calls are slowing page navigation.	• Reduce the number of REST calls that need to be made on each page. • Be careful not to create heavyweight back end triggers that can cause REST APIs to slow down.
There have been cases when multiple copies of jQuery are being loaded, unnecessarily.	Ensure that there are no files being loaded unnecessarily.

Potential Issue	Performance Tuning Tip

Service worker scripts cause performance degradation.	Disable service workers if they're enabled.
	
	You can completely disable the loading of the service worker script by adding the following to your index.html file:
	
	`<!-- visualBuilderScripts -->:`
	
	`<script type="text/javascript'>`
	
	`window.vbInitConfig.SERVICE_WORKER_CONFIG = {`
	
	`disabled: true`
	
	}
	
	`</script>`

When a record is created or updated performance is degraded with the back and forth from the server which calculates all fields each time.	Turn off unnecessary responses for POST and PATCH operations. Most create and update operations don't require a response from the server so the responsiveness of both the CSP and CSE applications could be improved by turning them off.
Slow performance	Limit the size of response payloads by using the fields= parameter in the request.
Slow performance	Use the `onlyData` parameter and set it to True.
Slow performance	Combine multiple API requests into single requests when possible • When many fields are needed in the response, use the expand= parameter on child settings. • Use the `fields=` parameter to return the children and grandchildren records with identified field of interest. • Use `Upsert-Mode: true` command to update or insert during POST to avoid multiple calls to check for an update versus an insert operation.
Slow performance	Configure bundle modules You can minimize what needs to be loaded on initial load by loading only the bundle that contains files needed on initial start up. For example, define a bundle for files needed for anonymous access, and separate bundle for later flows that require authenticated access. For more information, see the Related Topics for a link to the Bundle Modules topic from the Developing Applications with Visual Builder guide.
Access to secondary servers might be delayed by DNS resolution.	Use pre connect to establish early connections to important secondary servers, such as the Fusion Service and the CDN server if one is used, by adding a directive to index.html. For example, `<link rel="preconnect" href="https://example.com">`

Related Topics

- Optimize Your Application for Deployment

Optimize Your Application for Deployment

It's strongly recommended that you optimize your Digital Customer Service application for Production deployment.

You can optimize the performance of your published Digital Customer Service application by using node and Grunt to build them locally or to set up build jobs on Oracle Developer Cloud Service. For more information about optimizing your application, refer to the Optimize Your Builds topic in the Developing Applications with Oracle Visual Builder guide.

For more information on :ce values, refer to the Manage Your Component Exchange link in Related Topics.

The instructions listed here reflect a Windows deployment. If you are using Unix, change the path appropriately. For example: ./node_modules/.bin/grunt.

1. You must first install the `Node.js`. (npm is included in the Node.js installation).

   To download the appropriate installation package for your operating system, and for more information on the installation and configuration of it, refer to: `https://nodejs.org`.
2. Run the following one-time command from a Command Prompt in Windows or Shell window in Unix to install grunt CLI globally:

   ```
   npm install -g grunt
   ```
3. Create your Digital Customer Service application in the Visual Builder Designer.
4. Each time you want to optimize the VB app you've created, run the following:
   a. Export the VB app from VB Designer to the local machine.
   b. Unzip VB app to your local directory.
   c. From the Command Prompt, change the directory to the top level where the application is expanded and the `Gruntfile.js` file located.
   d. Install a local copy of grunt by running the following command: `npm install`
   e. Run the following processes using these examples:
      - `.\node_modules\.bin\grunt vb-clean`
      - `.\node_modules\.bin\grunt vb-process-local vb-package --url:ce=<component exchange URL> --username:ce=<component exchange username> --password:ce=<component exchange password>`
      - `.\node_modules\.bin\grunt vb-deploy --url=<https:<VBCS host>/ic/builder> --username=<VBCS username> -- password=<VBCS password> --id=<VBCS application ID> --ver=<VBCS app version> --profileId=user_proxy_prod --publish`

        > **Note:** To obtain the VBCS application ID open the app in the VBCS designer and see it in the address bar. For example, for the application `My App 1`, typically the following address is shown: `https://<host>:<port>/ic/builder/?root=application&application=My_App_1-1.0&artifact=Welcome` where `My_App_1` is the application ID, and `1.0` is the application version. Visual Builder design time replaces certain special characters in the application name such as space and hyphen with underscores to generate the application ID, so they are not always the same. Another way to get the application ID is from the published (live) application's URL. In our example, when the application is published, its URL would be: `https://<host>:<port>/ic/builder/rt/My_App_1/live/webApps/dcs/`, where `My_App_1` section of the URL is the application ID.

If your VB App is in Source Control in VB Studio, you can use the pipelines to automate the optimization tasks and reference.

Related Topics

- Manage Your Component Exchange
- Grunt Getting Started Guide
- Optimize Your Builds and Audit Your Code Using Grunt
- Create and Set Up the Project for Development
- Set Up the Project to Deploy for Production

14 Troubleshooting

Troubleshooting

The following table lists troubleshooting information.

Problem Description	Solution
There's an issue with the Google Chrome browser and user log in continually looping. Chrome has a 4k limit on cookie sizes. If the user has a very long first and last name that combination can cause the cookie that's generated to be larger than 4k. Chrome will only accept the first 4k of the cookie, making it incomplete. This results in the login process continually looping.Firefox doesn't have the 4k cookie limit and can be used to verify that this is the issue. For example the following details would cause the cookie to be too large: • First name: selfserviceuser@mycompany.com • Last name: selfserviceuser@mycompany.com • Email address: selfserviceuser@mycompany.com	Reduce the size of the First and last names. For example • First name: Self • Last name: user • Email address: selfserviceuser@mycompany.com **Note:** The email address can stay the same. This is performed in Identity Cloud Service by updating the user facing the issue.
After sign in, the VB page keeps reloading	One option is to allow third party cookies in the browser. The other solution is to make the following changes to the application which change the service connection type for Fusion REST APIs used with the Self Service Optimization flow: which changes the connection from Delegate Authentication to your Oracle Cloud account: 1. Open Visual Builder Designer and the application with the issue. 2. Open **Settings > Application Profiles** and edit the **fa server** for the User Proxy Production profile. 3. Set the authenticated user `auth` type to **Oracle Cloud Account**. 4. Open the **dcs web app > Settings > Security**. 5. Uncheck **Enable implicit grant for Service Connections**.

Related Topics

- How do I add Visual Builder roles?

Change the Appearance of your Digital Customer Service Application

This topic describes how to change the appearance of your own Digital Customer Service application.

The Oracle Digital Customer Service Reference Implementation template has been styled to enhance its appearance. This has been done by changing objects and adding styles to the `app.css` file.

To change objects in the `app.css` file:

1. Navigate to the Oracle Visual Builder.
2. Open your Digital Customer Service application.
3. Click **Web Apps**.
4. Expand `dcs`, expand `resources`, expand `css`, then click `app.css`.

   The `app.css` tab appears.
5. Find and configure the object that you want to change. For example, to set the header of the Reference Implementation template to a transparent black, change the **odcs-header** CSS class selector:

```
 .odcs-header {
 background-color: rgba(0, 0, 0, 0.7);
 height: 58px;
}
```

CSS Classes defined in `app.css` can then be referenced in the HTML of the application. For example we've this code in `pages/shell-page.html`:

```
<header role="banner" id="header" class="odcs-header oj-web-applayout-header>"
```

You can also use Oracle JET themes to provide consistent appearance of components across your Digital Customer Service application. For more information about Oracle JET themes see the `Theming Applications` chapter in the Developing Applications with Oracle JET guide, in the Related Topics.

Once an Oracle JET theme is created it can be added uploaded to `resources/css` and then referenced in the `index.html` of the application with code like this:

```
...
 <link type="text/css" rel="stylesheet" href="resources/css/app.css">
 <link type="text/css" rel="stylesheet" href="resources/css/myJETTheme.css">
...
```

Related Topics

- Theming Applications