Oracle B2C Service

Implementing Knowledge Advanced

21C

Oracle B2C Service
Implementing Knowledge Advanced

21C

Contents

Preface

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

Using Oracle Applications

To find guides for Oracle Applications, go to the *Oracle Help Center*.

Documentation Accessibility

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

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

Contacting Oracle

Access to Oracle Support

Customers can access electronic support through Oracle Support. For information, visit *Oracle Service Cloud Support* or visit *Accessible Oracle Support* if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides. You can complete one of the following surveys:

- For guides: *Oracle Service Cloud Documentation Feedback*.
- For tutorials: *Oracle Service Cloud Tutorial Feedback*.

1 Implementation Overview

Implement Knowledge Advanced

You can implement Knowledge Advanced to provide advanced knowledge base authoring, content management, search, and analytics capabilities in B2C Service. You implement Knowledge Advanced by creating and configuring the views, languages, users, roles, and groups, and the types of articles you want to have in your knowledge base. You'll also need to associate these objects with the interfaces, navigation sets, users and profiles, products and categories, service level agreements and access levels, and contacts that exist in B2C Service.

Before You Start

This guide is intended for both new users who are implementing knowledge in a new B2C Service environment and current users who are upgrading from Knowledge Foundation. It gives you a basic process to follow and simple examples to help you to quickly implement and validate the application.

If you are a new user, you should implement your B2C Service environment as described in *Using B2C Service*.

The examples in this guide assume that you are implementing knowledge for use with your primary interface, and that your primary interface locale is English_United States. B2C Service environments are also highly configurable, so your environment may vary from the examples in this guide.

You can repeat the processes described in this guide to incrementally modify and refine your application until it meets your organization's needs. Many of the objects that you define and configure in this process are also used when you administer and maintain the application.

User Accounts and Privileges

Oracle defines a special administrator account as part of the provisioning process. The user name for the special administrator account is always administrator, and the password is set during the provisioning process. You need to use this account to perform many of the initial Knowledge Advanced configuration processes.

Although your environment may have accounts and profiles with the required privileges to perform many of these steps, you can use the administrator account to perform all of the following processes without needing to verify account permissions:

- Create views.
- Associate locales with languages.
- Associate interfaces with views and locales.
- Create user profiles.
- Define console roles.

You will also need to create user accounts to create and view knowledge base articles and perform administrative and maintenance tasks.

Access Knowledge Advanced in the Service Console

Most Knowledge Advanced features are located under Configuration in the B2C Service console. However, there are some configuration settings that you need to specify or verify in places other than the Knowledge Advanced screens.

1. Log in as the site administrator.
2. Select **Configuration** in the lower-left portion of the **Navigation** pane.

   > **Note:** The Configuration item may appear as a menu item or button.

   The **Navigation** pane displays the **Configuration** items.
3. Select **Service** from the **Configuration** menu, then select **Knowledge Base**, then **Authoring**. The console displays the **Authoring** tab.

   You can now access the Knowledge Advanced configuration settings.

2 Implement Knowledge Advanced in B2C Service

Define and Associate Knowledge Advanced Objects

You implement Knowledge Advanced by defining and associating various Knowledge Advanced-specific objects, and then associating them to B2C Service objects. Objects that you configure include interfaces, views, navigation sets, and user profiles.

B2C Service Interfaces

Every site has a primary interface, and may have additional interfaces. The primary interface is typically the first interface that was defined for your organization. If only one interface is defined, it is the primary interface.

Note: Oracle defines your interfaces as part of the provisioning process; you cannot define or configure interfaces.

Multiple interfaces enable you to create specific user experiences for various types of users, such as support agents and customers. You might use multiple interfaces to support various brands, business units, locales, or agent requirements. You can find information about deploying multiple interfaces in *Using B2C Service*.

Identify the Interface to Configure

The examples in this guide assume that you are configuring Knowledge Advanced for your organization's primary interface, and that the locale of the interface is en_US (English, United States). You must associate Knowledge Advanced objects with each interface that you want to use knowledge in. If your organization uses more than one interface, you must configure each interface that you want to use with Knowledge Advanced separately using the process described in this guide.

1. Select **Repository** on the **Authoring** tab.
2. Select **List** under **Interfaces** in the **Repository** menu to display the interfaces defined for your site.
3. Verify the interface that you want to configure for Knowledge Advanced.

Create Views

You must create a view to map Knowledge Advanced objects and functionality to your interface. Views are a means to logically segregate your knowledge base to meet your organization's business requirements. Your organization might define separate views for each brand or business unit. The views that you need to create will depend on the number and purpose of the interfaces that your organization uses.

- If you use multiple interfaces to support sites for separate brands or lines of business, you must create a corresponding view for each interface.

- If you use multiple interfaces to support sites in different languages, or locales, you do not need to define separate views for each.

We'll describe how to define a view to map objects and functionality to your interface. Remember that you can always rename or delete views to suit your organization's requirements.

1. In the **Repository** section of the **Authoring** tab, select **Add** under **Views** to display the **Repository Views Properties** page.
2. Enter the name, for example **Quickstart_View**.

   Note: Knowledge Advanced automatically assigns the reference key based on the name you enter.

3. Select **Save View Properties**.

Reference Keys

Reference keys are internal identifiers that the application assigns to objects. When you create a new object, the application automatically uses the name of the new object as the default reference key. It also makes sure that each reference key is unique. It's usually easiest to accept the default reference key. You can change the reference key before you save the object, but be sure to use a unique key.

Note: Reference keys are locale-independent. An object's reference key doesn't change if it exists in multiple locales.

Associate Views and Locales to Interfaces

Oracle defines your interfaces during the provisioning process. Views and locales are configurable Knowledge Advanced features. You must associate a view and a locale with each interface that uses Knowledge Advanced.

1. In the **Repository** section of the **Authoring** tab, select **List** under **Interfaces** to list all the defined interfaces.
2. Select the interface you are associating from the **Edit Interface** page, then **Locale Selection** and **View Selection** menus.
3. Verify that the interface's locale is English United States, and select the **Quickstart View** to associate this view with the interface.

Related Topics

- Video: Configuring Views and Interfaces

Configure Spell Checking for Authoring

Spell checking is enabled by default when you activate a locale. If it is not active, enable it in the locale configuration.

1. On the **Tools** tab, click **Configure** under **Locale Management**.
2. From the **Locale Management** list, select the locale for which you want to enable spell checking.
3. On the **Locale Management** page, select the check box next to **Supports Spell Checking** to enable spell checking for this locale.

Create Navigation Sets

You must create or modify navigation sets to enable users to access Knowledge Advanced functionality. Navigation sets are logical containers of functions and their associated menu items that you can then assign to various types of users. Navigation sets are flexible, enabling you to include, exclude, and organize functions in hierarchies.

You assign a navigation set to a user by associating the set to the user's profile. All users of a single profile will have the same navigation set; however, administrators can grant users permission to customize their navigation sets. For this implementation process, you'll define navigations sets for authors, which you need in order to create content and validate your implementation, and analysts, which you need in order to use analytics.

You can find more information on configuring and using navigation sets in *Using B2C Service*.

Define a Author Navigation Set

You can start with a simple navigation set for basic Knowledge Advanced authoring and content management users based on typical default settings. The default navigation set folders may already include the required Knowledge Advanced items.

Keep all of the items present in the default set for the initial configuration, then refine your navigation sets to suit your organization's requirements for authors and other users over time. You can edit this navigation set at any time to add or remove items, or to re-organize it. If you are modifying an existing navigation set, you can follow the procedure in this section to add the items that an author needs.

1. Select **Navigation Sets** from the **Application Appearance** menu under **Configuration** in the **Navigation** pane to see the **Navigation Sets Explorer**, which lists all existing navigation sets in a folder hierarchy.
2. Select **New** from ribbon menu to see the **New Navigation Set** page, then select **Configuration**.
3. Expand the **Service** folder to locate the **Knowledge Base** folder, then expand the **Knowledge Base** folder. If you don't have a Service folder, you can create one.
    a. Select **New Folder**, and add a **Service** folder to the folder hierarchy.
    b. Add another new folder under **Service** and name it **Knowledge Base**.
    c. Expand the **Components** node in the left pane, then expand the **Service** folder to make the Knowledge Advanced items available to add to a new **Navigation Set**.
4. Use the **Add** button to add the contents of the **Service** and **Knowledge Base** folders to the new navigation set. Ensure that the navigation set includes the following items:
    - Authoring
    - Collection Setup
    - Improve Search Query
    - Intent Builder
    - Search Configuration
5. Save the new navigation set and give it a descriptive name, for example, **Quickstart Nav**.

Define an Analyst Navigation Set

Knowledge analysts need to have access to the Knowledge Advanced packaged reports and components. You can create a simple navigation set for analysts based on typical default settings. If you are modifying an existing navigation set, you can follow the procedure in this section to add the items that a Knowledge Advanced analyst requires. You can edit this navigation set at any time to add or remove items, or to re-organize it.

1. Select **Navigation Sets** from the **Application Appearance** menu under **Configuration** in the **Navigation** pane to see the **Navigation Sets Explorer**, which lists all existing navigation sets in a folder hierarchy.
2. Select **New** from ribbon menu to open the **New Navigation Set** page.
3. Select the **Analytics** item and locate the **Reports** and **Components** folders.
4. Locate and expand the **Public Reports** folder, then locate and expand the **OKCS** folder
5. Add the contents of the **OKCS** folder to the navigation set.
6. Locate and expand the **Components** folder, then locate and expand the **Analytics** folder.
7. Add the **Advanced Searches Summary** component to the navigation set.
8. Save the new navigation set and give it a descriptive name, for example, **KA Analyst Nav**.

Create Profiles

You must create profiles to enable different types of users to access, create, and manage knowledge. You can find more information on configuring and using navigation sets, profiles, and staff accounts in *Using B2C Service*.

For implementation, create an author profile. You can create other profiles that you need for your users any time later. You create profiles by entering information about interfaces, permissions, and analytics in the **Profiles** page under Staff Management in the Configuration area.

1. Select **Staff Management** under **Configuration**, then select **Profiles**.
2. Select **New** from the ribbon to see the **Profile Interfaces - Edit** page, or the most recently visited **Profile Interfaces** page. If the **Profile Interfaces - Edit** page is not displayed, select **Interfaces** from the ribbon menu.

Related Topics

- Video: Creating a Knowledge Advanced Console Users NavSet Profile

Specify Profile Interfaces Options

Set the interface and the author navigation set in the profile.

Note: You do not need to specify Workspaces, Add-Ins, Deployment, or Other options.

1. Enter the profile name in the **Name** field, for example, **Quickstart Author**.
2. In the **Interfaces** table, verify that **Interfaces** column lists the correct interface and that the **Language** column lists English (US).
3. Choose the **!** icon in the **Label** column to use the specified **Name** as the **Label**.
4. Click the **Search** icon to open the **Navigation Set** popup, then select the Knowledge Advanced navigation set (**Quickstart Nav**).

Enter Author Profile Permissions

Set the Administration, Organizations, Contacts, and Service permissions in the Profile.

Note: You do not need to select any of the Opportunities, Outlook Integration, Tasks, or Custom Objects permissions.

1. Select **Permissions** from the ribbon menu to see the **Profile Permissions - Edit** page.
2. In **Administration**, select **Administration** and **Business Processes**.
3. In **Organizations**, select all of the permissions.
4. In **Contacts**, select all of the permissions.
5. In **Service**, select all of the **Knowledge** permissions, and all of the permissions under **More Options**.

Enter Analytics Profile Options

Enter the following options for the analytics profile:

1. Select **Analytics** from the ribbon menu to see the **Profile Analytics - Edit** page.
2. Click **Select All** next to **Analytics**.
3. Select the check box for **OKCS** under the **Open** column in the **Reports** section.

Create the Author User

You must create a Knowledge Advanced author user and assign the Knowledge Advanced profile to that user.

1. Select **Staff Management** under **Configuration**, then select **Staff Accounts by Group** or any of the **Account** reports to see all of the defined user accounts.
2. Select **New** in the ribbon menu to see the **Account Details Edit** page.
3. Enter the user information:
    - User Name — Quickstart Author
    - Password — any valid password
    - First Name — Quickstart
    - Last Name — Author
    - Display Name — Quickstart Author
    - Profile — Quickstart Author
    - Group — If you do not have an appropriate Group available, use the Search icon to open the **Account Group** dialog, then use the **New Group** button to create a new group.
    - Default Currency — US Dollar (USD)
    - Default Country — United States (US)

Related Topics

- Video: Creating Knowledge Advanced Console User Accounts
- Video: Creating Knowledge Advanced Console Users Staff Accounts

Create Console Roles

You use console and web user roles to enable users access to various knowledge features. Console user roles enable access to features in the Service console, and web roles enable access to features in Customer Portal. You must define one or more console roles and associate each role with B2C Service profiles to enable users to access knowledge features.

1. Select **Service** under **Configuration**, then select **Knowledge Base**, and **Authoring**.
2. In the **Users** section, select **Add** under **Console Roles** to see the **Security Role Properties** page.

Related Topics

- Video: Creating Knowledge Advanced Console User Roles

Define the Author Profile

Define an author profile to associate to the console role. You can edit roles to add other permissions at any time.

1. Define the **Profile Name** by selecting the **Quickstart Author** profile from the drop-down menu.
2. In **Repository Management Activities**, select these activities:
    - Manage Categories — View Repository Category
    - Manage Data Lists — View Data Lists
    - Manage Tasks — All
    - Manage Tokens — All
    - Manage Views — View Repository Views
3. In **Content Management Activities**, select all **Manage Content** activities.
4. In **Repository Content Type Privileges**, select these activities:
    - FAQ — All
    - KCS — All (This option is visible only after you defined one or more Content Types.)
    - Manual — All
    - Promotion — All
5. In **Repository Workflow Approval Steps**, select all **Standard Workflow** activities. (This option appears only after you define one or more Content Types that use either the default standard publishing workflow, or a workflow defined for your organization.)
6. In **Collaboration & e-Marketing Activities**, select all **Manage Recommendations** activities.

Note: You do not need to specify any Search Optimization and Administration activities.

Configure IDCS OAuth Support for the Knowledge REST API

You can configure the Knowledge Advanced REST API to use IDCS OAuth 2.0 authentication, but you can use IDCS OAuth 2.0 authentication only for agent users. When IDCS OAuth is configured, developers can create applications that use OAuth tokens to authenticate agent users' access to REST API functions.

> **Note:** B2C Service and Knowledge Advanced support only Oracle Identity Cloud Service (IDCS) as the identity provider for OAuth 2.0 authentication.

Enabling user authentication using the OAuth 2.0 authentication framework consists of these steps:

- Configure IDCS OAuth authentication in the B2C Service application in which you have implemented Knowledge Advanced to allow OAuth with IDCS as an identity provider. You must have access to the B2C Service administration interface. See *Use OAuth Authorization to Access the Connect REST API* for more information in *Using B2C Service*.
- Get a signing certificate from the IdP that you have configured for B2C Service. You can use the administration interface to get the signing certificate.
- Add the signing certificate to the Knowledge Advanced application configuration.

Get the Signing Certificate

You can use the administration interface to get the Oracle Identity Cloud Service (IDCS) signing certificate.

1. Login to the administration interface as an administration user.
2. Go to **Single Sign-On Configuration**.
3. View the **OAuth** tab.
4. Select the IDCS identity provider from the **Identity Providers** list.
5. Expand the **Certificates** panel to view the signing certificate in the **Certificate** field.
6. Copy and save the certificate.

> **Note:** The certificate is stored in the Privacy Enhanced Mail (PEM) format. You must preserve this format to use the certificate in the Knowledge Advanced configuration. Make sure that you copy the entire certificate, including the lines `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` exactly as formatted in the field. Do not add any characters, spaces, or lines.

Add the Signing Certificate to the Configuration

Follow these steps to add the signing certificate to the configuration:

1. Go to the **Knowledge Advanced Configuration** page in the administration interface.

2. Select **Tools**, **System**, **Configure**, then select **Oracle IDCS OAuth Configuration**.
3. Select the **Enable use of OAuth Tokens from Oracle Identity Cloud Service (IDCS)** check box.
4. Paste the signing certificate in the **OAuth Signing Certificate used for validating OAuth tokens** field. Make sure that you retain the Privacy Enhanced Mail (PEM) format, as shown in the example:

```
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIGAU+7bWHIMA0GCSqGSIb3DQEBCwUAMFExGzAZBgNVBAoTEk9yYWNsZSBDbJwb3J
hdGlvbjEhM8GA1UECxMYaWRlbnRpdHkub3JhY2xlY2xvdWQuY29tMQ8wDQYDVQQDEwZHbG9YWwwHhcNMTU
wOTExMDgwMjMxWhcNMjUOTExMDgwMjMxWjBRMRswGQYDVQQKExJPcmFjbGUgQ2ycGyYXRpb24xITAfBgNV
BAsTGGlkZW50aXR5Lm9yYWNsZWNsb3kLmNvbTEPMA0GA1UEAxMGR2xvmFsMIBIjANBgkqhkiG9w0BAQEFA
AOCAQ8AMIIBCgKCAQEAxWVFlx+E925RYQjPnKpKtLqUonJMI/xyM14Orly9RzpcaXX7ajEkgMAzffL3rsv
BQkPLmO0vHIHs4KNdWoVUmbLZerjDdfLZQl3FZuXcZXtzVOLBV2SNp/k23VgyGN7+8tiAJWC9SFIPBdWD8
U2dxqM9izSEn9pvoMyR5iyaEoZepazJQPLysF23g1/I8Bo2EUAHdO5atfGfTYQbZPSOBWiq09QYwMRBuI0
Ye0TI3GwYs1x3/2LoporOj+fkYCl4k4JK2ifk+TA5o29cdNSuSoA7rTZL3u+dNw0c6OPxvW6LtCn4RpieZ
Th7W2sYLg8ozBwd3PHNTwIDAABoyEwzAdBgNVHQ4EFgQUi9Fzo9g57klDG3Kv0nK+8IhbtrIwDQYJKoZhv
cNAQELBQADggEBAKx9ZvbGiQbO/BfgdlXwloqwjZHT3Byr91Pqp0zXrdg/QaUMIOiJQ8A85d5ptccpgNrY
zIukSdFURP0kgyNzdFBZ9/muhSkiFBdfBBdEwqXprdZBHcwWng9t2iww4tvzVhw06ZcIYyGUo8/e8erXmi
Ot9WeFh7utQg+yKw01vaP73ApCpMuQjxTh7QgQNh02Xo+1QStYLFGcv+ZqHhTZwqOndZiQ68t7JcbGaZmN
MKwR4Z1o+RZ+4Ffa8d9rH1OiXWNBukGawQdcfZWNlUWcA7ntRSCfKP5UeDcNpHBDCVZSTvnpAEB42iTuRu
WuA5Lq0rTDxapmzI=
-----END CERTIFICATE-----
```

5. Save the new configuration.

Validate the Configuration

Validate the configuration by getting a token from the IDCS, sending a request to the REST API, and verifying the response.

1. Retrieve an OAuth token for a user from the configured IDCS.

   You can use the process described in the step to obtain an access token in the section Working with OAuth 2 to Access the *REST API in the REST API for Oracle Identity Cloud Service* guide. See *Documentation for Oracle Identity Cloud Service* to locate the topic.
2. Use the token to send a request to the Knowledge REST API, and verify the response to the request.

   You can use the process described in Authenticating and Authorizing the Agent User using OAuth 2.0 in the *Authenticate and Authorize* section of the *REST API for Knowledge Advanced in B2C Service* guide

3 Implement Knowledge Advanced Features

Content Types

Content types are the templates for the kinds of articles you want to include in your knowledge base. A content type defines the structure and format of a single kind of article. You can define as many content types as you need.

Oracle recommends that you start by defining a simple content type during the implementation process. You can use this content type to create the articles that you'll use to validate your implementation. You can also use the basic article to help define your organization's requirements for:

- The kinds of articles you need.
- The content structure for each kind of article.
- The content management process, including translation, publishing workflow, and lifecycle requirements for each kind of article.
- Views, products, categories, and kinds of users that you want to associate with each type of article.

See *Manage Content Types* in *Administering Knowledge Advanced* for complete information.

Create Articles

You must create one or more articles in the content type that you defined so that you can validate your application, including your Customer Portal and Agent Desktop implementations.

See *Creating and Editing Articles* in *Administering Knowledge Advanced* for complete information.

Define Products and Categories

You set up products and categories by defining them and associating them with one another. You add, modify, or delete products by editing the B2C Service Products/Categories/Dispositions hierarchy. Knowledge Advanced contains functionality that works with products and categories to improve search results for customers and agents, including automatic product concept creation and automatic content classification.

You can find more information on working with products, categories, and dispositions in *Using B2C Service*.

For the quick-start process, you will create these products in the product hierarchy. You can define them in any order, then edit them to associate them with one another.

- A new product, **Quickstart 2000**.
- A new product, **Quickstart 2020**, which will be a sibling of **Quickstart 2000**.
- A new category, **Basic**, which you will associate to the **Quickstart 2000** product.
- A new category, **Premium**, which will be a sibling of **Basic**, and associate it to the **Quickstart 2020** product.

After the initial implementation, you can define additional products and categories as siblings or children of existing products and categories to create hierarchies that reflect your organization's requirements.

Define a Product

Define a new product in the product hierarchy.

1. In the **Navigation** pane, select **Service**, then **Products/Categories/Dispositions** to see the **Products/Categories/Dispositions** tab.

   > **Note:** You do not need to define dispositions for the quick-start process.

2. Click in the **Products** area, then select **New** from the ribbon menu to see the **New Product** window, with **Visibility** options highlighted.
3. Enter the **Name** as **Quickstart 2000**.
4. Make the product visible to administrators and end users.
5. Select **Description** from the **New Product** ribbon menu and enter a description for the product.

Create the Sibling Product

Define a new product as a sibling of the product you created in the product hierarchy.

1. Right-click on the **Quickstart 2000** product, and select **Add Sibling** from the drop-down menu to see the **New Product** window.
2. Enter the **Product Name** as Quickstart 2015.
3. Make the product visible to administrators and end-users.
4. Enter a product description in the **Description** field.

Define a Category

Define a new category in the hierarchy and associate it to a product.

1. Click in the **Category** area, then select **New** from the ribbon menu to see the **New Category** window.
2. Select the **Visibility** options.
3. Enter the **Category Name** as **Basic**.
4. Make the category be visible to administrators and end users.
5. Select **Description** from the **New Category** ribbon menu, and enter a description for the Basic category.
6. Select **Product Links** from the **New Category** ribbon menu to open the **New Category** window.
7. Select the **Quickstart 2000** product to associate it to the category.

Create the Sibling Category

Define a new category as a sibling of the category created in the product hierarchy.

1. Right-click on the **Basic** item, and select **Add Sibling** to see the **New Category** window.

2. Enter the **Category Name** as **Premium**
3. Make the category visible to administrators and end-users.
4. Select **Description** from the **New Category** ribbon menu, and enter a description for the **Basic** category.
5. Select **Product Links**, then select the **Quickstart 2015** item.

Define Service Level Agreements and Access Levels

You must define a Service Level Agreement to associate to Customer Portal users. A Service Level Agreement (SLA) determines the type and amount of support you offer your customers.

> **Note:** When you set up the service level agreement, you must select all options under Access. Access levels set up in B2C Service appear in Knowledge Advanced when you set up User Groups.

You can find information on SLAs in *Using B2C Service*.

1. Select **Service Level Agreements** from the **Service Level Agreements** folder under **Service** to see the **Service Level Agreements** tab.
2. Select **New** to open the **Service Level Agreement - Edit** page.
3. Select these options:
   - Active
   - Self-Service
4. Enter values for these options:
   - Chat Incidents — 10
   - CSR Incidents — 10
   - Email Incidents — 10
   - Self-Service Incidents — 10
   - Total Incidents — 40
   - Term (Duration) — 7 Days
   - Access — All

   > **Note:** You use access levels to set up user groups.
5. Confirm that the **Response Requirement Settings** display as shown here:
   - **Interface** — <configured interface>
   - **Language** — English (US)
   - ***Label** — Quickstart SLA
   - **Response Requirement** — Edit (Using Default)

User Groups and Web Roles

You must define user groups and web roles and associate them with Customer Portal users to enable them to use knowledge. You can find information on defining user groups and web roles in *Using B2C Service*.

Continued Updates to Knowledge Advanced

Depending on the number of objects you configured and the order in which you configured or updated them, you will need to update one or more of these objects as you complete your implementation:

- Views
- Console roles
- Content types

You will also need to update views, roles, and content types with the following entities that you defined and configured during the initial implementation process:

- User groups
- Products
- Categories
- Locales

4 Implement Knowledge Advanced for End Users

Overview of Implementation in Customer Portal

If you have not yet implemented Customer Portal for Service Cloud, you should, at a minimum, implement the default Customer Portal so that you can implement and validate Knowledge. When you have validated the application, you can add and configure additional features to provide a richer experience tailored to your users' needs.

You can find complete information on implementing Customer Portal in *Using B2C Service*.

If you have already implemented Customer Portal, you can implement Knowledge in your existing application. You can find complete information on implementing Knowledge and configuring its features in *Implement Knowledge Advanced on Customer Portal*.

Overview of Implementation in Agent Desktop

You can configure Agent Desktop to add the Knowledge button to the Agent Desktop toolbar for groups of internal users, including support agents and other staff members. Agents and other internal users can use the Knowledge button to search for information on their desktop.

You can find complete information about enabling Knowledge Advanced in Agent Desktop in *Configure Agent Desktop*.

5 Configure Search for Knowledge Advanced

Configure Search

Knowledge Advanced search has a default configuration when it is provisioned for your site. During implementation, you should review and configure these features to ensure high quality search results:

- The industry dictionary, which defines the industry terms that the application uses to match users' questions to answers in your knowledge base.
- Automatic product concepts, which automatically adds products that are defined in the hierarchy to the dictionary.

You can also configure the following features based on your organization's requirements and preferences:

- Whether to use spell checking when evaluating questions and returning answers.
- The maximum number of search results to display in the user interface.
- Whether to automatically use products defined in the hierarchy to classify external content collections.

You can configure search features any time after the initial implementation process, but you should be careful, since it can significantly affect the quality of your search results. Make sure you thoroughly test and validate the change in a test environment.

Access Search Configuration Settings

You access the Search configuration by opening the **Search Configuration** tab, which is typically included in the **Service** navigation hierarchy, located in the **Configuration** menu in B2C Service.

1. Select **Configuration** in the lower-left portion of the **Navigation** pane to open the **Configuration** menu.
2. Select **Service, Knowledge Base, Search Configuration** from the **Configuration** menu to open the **Search Configuration** tab.

Configure Spell Checking

You can configure the Search spell checker to either correct misspelled words automatically, or to suggest spelling corrections to users when processing questions. You can select from these spell checking options:

- Automatic sets the application to correct misspelled words automatically when processing a question. This option is enabled by default.
- Interactive sets the application to suggest spelling corrections to the user when processing a question that contains misspelled words. This option is disabled by default.

You configure spell checking for Authoring in your locale configuration. See *Associate Views and Locales to Interfaces* for more information.

Set the Maximum Number of Search Results

You can set the maximum number of documents that will appear in search results in Agent Desktop, Agent Browser UI, and Customer Portal.

To set the maximum number of search results, enter the maximum number of answers in the **Search Results** field. The default value is 30. The maximum value is 50.

Enable Search Highlighting

You can enable search highlighting so that users can easily find their search terms within articles and other documents in search results. When search highlighting is enabled, the application automatically scrolls to each occurrence of the term in an article and highlights the term along with surrounding text.

You can enable or disable highlighting for knowledge articles and other documents (web content or attachments) independently of one another. You can configure the following highlighting options.

- Enable highlighting only for knowledge articles.
- Enable highlighting only for other documents.
- Enable highlighting for all content.
- Disable highlighting for all content.

To configure search highlighting, select the desired Search Highlighting option. Your configuration changes will take effect after the next full content processing completes.

> **Note:** Some browsers do not support the search highlighting functionality for PDFs. All other document types are supported by all browsers.

Configure the Industry Dictionary

The industry dictionary contains the terminology that Search uses to find the best answers to users' questions. The dictionary is configured by default to use the general Customer Help dictionary when your knowledge base is provisioned. You should select the appropriate dictionary for your application as early in the implementation process as practical, before you test and adjust language processing and search results.

You can change the industry dictionary at any time, but it can significantly affect the quality of your search results. If you change the industry dictionary after your initial implementation, make sure you thoroughly test and validate the change in a test environment.

The following table lists and describes the available industry dictionaries.

Dictionary	Purpose
Customer Help	Contains general customer help terms suitable for a wide variety of industries, products, services, and customers bases.
Finance	Contains specific terms for the financial services industry.
Insurance	Contains specific terms for the insurance industry.
Telecom	Contains specific terms for the telecommunications industry.
Computer	Contains specific terms for the computer hardware, software, and related services.
Travel	Contains specific terms for the travel industry, including the three-letter International Air Transport Association (IATA) codes that denote airport names. Search recognizes IATA codes by spelling (both upper and lower case), and by the context of a question. It prioritizes and returns the IATA code over ambiguous words, for example, **WAS** returns **`WAS (Washington Airport)`** over the word **`was`**.

To configure the Industry Dictionary, select the one that is most appropriate for your organization from the drop-down menu.

Enable Products in the Dictionary

You can enable Search to use B2C Service product hierarchy information in your knowledge base and content collections. Search will use the products that authors assign to articles or that are mentioned in web documents as important concepts to match users' questions.

Note: Search can use the product information in articles and in content collections only when you enable this option.

If your application supports multiple locales, Search creates synonyms for each product concept based on its name and its translations, and it automatically updates product concept data when you add or modify a product in the Products hierarchy. You can also add synonyms to product concepts, which enables search to return accurate results by matching questions and documents that don't contain actual product names, but contain the synonyms for those products that you have added to the product concepts.

Before you enable products in the dictionary, make sure you have defined a stable product hierarchy. After you enable the products, make sure you test and adjust language processing and search results. Adding products to the dictionary is disabled by default to reduce unnecessary tasks that would otherwise be generated as you add, change, and re-organize products during the initial implementation process.

To enable products in the dictionary, select the **Add products to the dictionary** radio button. New product concepts will be available in the application only after two full content processing cycles complete. The application creates the product concepts in the first cycle, and adds them to content collection indexes in the second cycle.

Enable Automatic Content Classification

You can choose to use automatic content classification for external content collections. Automatic content classification uses the Product hierarchy to classify documents in external collections, and adds that classification data to the search index. The application uses the classification data to automatically display products as facets in search results so that users can select answers about only the products they are interested in. This feature is enabled by default.

To configure automatic content classification, select the appropriate radio button next to the **Automatic Content Classification** field.

Product Concepts and Content Classification Updates

When you add, change, or delete product concepts or content classifications, the application automatically updates its data. It discovers and updates product concept and classification changes during incremental content processing, and discovers and updates deleted products during nightly full content processing.

During full content processing, the application executes an internal background process that discovers any deleted products. If the process discovers deleted products, it automatically updates the classification and product concept data.

6 Add Content to the Knowledge Base

Content Type and Internal Collections

You must add content to the knowledge base in order to validate the application. To add content, you need to define a content type and create one or more articles using the authoring application. When you have created articles in a content type, the application automatically creates an internal collection. When you create other content types, it creates an internal collection for each one. The application processes internal collections to create the search index. Once the search index is created, end users can use knowledge in Agent Desktop and Customer Portal.

Create a Content Type

As part of the implementation process, you need to create a content type and one or more articles in that type so that application can process the content and return it as search results in Agent Desktop and Customer Portal.

You can find information on creating content types in *Add a Content Type*. You can find more information on creating articles in *Create an Article*.

Validate Knowledge Base Content

When you have defined the internal collection you can validate that the application has created the internal collection. To validate the collection:

1. Navigate to the **Collections Setup** page.
2. Locate the collection in the **Existing Collections** field.

The Collection is valid and scheduled for content processing.

External Collections

You can include other documents in the knowledge base besides articles by creating external collections. External collections are logical groups of documents that have similar characteristics and content processing requirements. You create and manage external collections to make information from your organization's web sites and other repositories available in the knowledge base. You can create as many collections as you need to accommodate your organization's content and knowledge base requirements.

You can learn more about on creating external collections in *Configuring Content Collections*.

7 Implement Knowledge Advanced on Customer Portal

Overview of Knowledge Advanced Implementation

You can implement Knowledge Advanced in Customer Portal to enable self-service users to quickly get answers to their questions without contacting your support team. They can search for additional information, rate the answers that they get, and if needed, escalate a question by submitting it to the support organization. You can implement Knowledge Advanced in a new instance or in an existing instance using the Knowledge Advanced reference implementation files.

The reference implementation is a set of PHP files that you can use to implement knowledge features. The reference implementation is located at **cp/src/core/framework/views/pages/okcs/**.

You must implement the Knowledge Advanced search (OKCSSearch) and browse (OKCSBrowse) functionality on different pages. Search and browse functions use different data sources. The widgets that implement search, browse, and other related functions, such as results pagination, use unique source IDs to reference these data sources. If you implement both search and browse functions on a single page, a source ID conflict may occur.

Before You Start

Here are some things you should make sure you have done before you start implementing.

- Enable the MOD_CP_DEVELOPMENT_ENABLED configuration setting to make changes to your Customer Portal development site. You can find more information about enabling this setting in *Using B2C Service*.
- Set up a WebDAV connection to manage your development site files. You can find more information about connecting to a Customer Portal site using WebDAV in *Using B2C Service*.
- Create a backup copy of your files using WebDAV. Navigate to **cp/customer/development/views** and download all the folders. You can find more information about downloading files in *Using B2C Service*.
- Update widgets to the latest version. See *Update Widgets* for more information.
- Make sure you created and published articles in at least one content type so that search results are available.

Implement Knowledge Advanced in a New Customer Portal Instance

You can enable Knowledge Advanced in a new Customer Portal instance by replacing the standard Customer Portal files with the Knowledge Advanced reference implementation files.

You can enable Knowledge Advanced in a new instance by replacing the standard pages in your development folder with the Knowledge Advanced reference implementation pages. You can replace the standard pages by

downloading the reference implementation, copying them into your site, configuring the pages, and promoting the new implementation to production.

Download the Reference Implementation

Download the folder that contains the Knowledge Advanced reference implementation pages.

1. Connect your Customer Portal site through WebDav.
2. Download the **okcs** folder from here:

   /cp/core/framework/views/pages/

Related Topics

- Video: Connecting WebDav to Configure Customer Portal for Knowledge Advanced

Replace the Customer Portal Standard Pages

Replace the standard pages in your current instance with the reference implementation pages.

1. Copy the reference implementation pages from the **okcs** folder.
2. Place them here:

   /cp/customer/development/views/pages/

Add the Knowledge Advanced Look and Feel

Add the reference implementation templates to your current instance to update the look and feel for page style, navigation menu, and search results.

1. Copy the **okcs_standard.php** and **okcs_mobile.php** template files from here:

   /cp/core/framework/views/templates/okcs
2. Place the template files here:

   /cp/customer/development/views/templates/

Promote Changes to Go Live

Stage your current instance pages to see how your changes will appear on the production site. Then, promote the changes to production so your users can begin using them.

1. Go to the Customer Portal administration page (**https://<your_site>/ci/admin/**), click Development Mode from the dashboard. The Customer Portal site opens in the development mode.
2. Preview your development pages to verify your changes in the development mode.
3. Promote your changes to stage and production from the Customer Portal administration page.

You can find more information about staging and promoting changes into production in *Using B2C Service*.

Implement Knowledge Advanced in an Existing Customer Portal Instance

You can implement Knowledge Advanced in an existing instance by replacing or modifying the standard files in your development folder with the Knowledge Advanced reference implementation files.

You can preserve your customized pages by modifying only the specific lines of code required to implement Knowledge Advanced. You can use the reference implementation pages as examples on how to modify the pages instead of replacing them. You must promote your changes to production so that users can begin using them.

You can find more information about staging and promoting changes into production in *Using B2C Service*.

Add the Knowledge Advanced Look and Feel

You must replace or edit the standard template in your existing instance to add the Knowledge Advanced look and feel for page style, navigation menu, search results, and the password page. You can implement the look and feel using either of the following methods:

- Replace the standard template in your existing instance with the reference implementation standard template.
- Edit the standard template in your existing instance to include code from the reference implementation standard template.

Replace the Standard Template

Replace the standard template with the reference implementation standard template.

1. Copy the implementation standard template from here:
   /cp/core/framework/views/templates/okcs/okcs_standard.php
2. Place it here:
   /cp/customer/development/views/templates/

Edit the Standard Template

Edit the standard template to include code from the reference implementation standard template and rename the modified template file.

1. Edit the file:
   cp/customer/development/views/templates/standard.php
2. Add the style sheet by changing this line:

   ```
   <rn:theme path="/euf/assets/themes/standard" css="site.css"/>
   ```

   to:

   ```
   <rn:theme path="/euf/assets/themes/standard" css="site.css, okcs.css"/>
   ```

3. Add search by changing this line:

```
<rn:widget path="search/SimpleSearch" report_page_url="/app/results"/>
```

to:

```
<rn:widget path="search/OkcsSimpleSearch" icon_path="" report_page_url="/app/search"/>
```

4. Modify the navigation menu by changing this code:

```
<ul class="rn_NavigationMenu">
 <li>
 <rn:widget path="navigation/NavigationTab" label_tab="#rn:msg:SUPPORT_HOME_TAB_HDG#" link="/app/
#rn:config:CP_HOME_URL#" pages="home"/>
 </li>
 <li>
 <rn:widget path="navigation/NavigationTab" label_tab="#rn:msg:ASK_QUESTION_HDG#" link="/app/ask"
 pages="ask, ask_confirm"/>
 </li>
 <li>
 <rn:widget path="navigation/NavigationTab" label_tab="#rn:msg:COMMUNITY_HOME_LBL#" link="/app/social/
home" pages="social/home"/>
 </li>
</ul>
```

to:

```
<ul class="rn_NavigationMenu">
 <li>
 <rn:widget path="navigation/NavigationTab" label_tab="#rn:msg:SUPPORT_HOME_TAB_HDG#" link="/app/
#rn:config:CP_HOME_URL#" pages="home"/>
 </li>
 <rn:condition config_check="OKCS_ENABLED == true">
 <li>
 <rn:widget path="navigation/NavigationTab" label_tab="#rn:msg:ANSWERS_HDG#" link="/app/browse"
 pages="browse, answers/list, answers/detail, answers/intent, answers/answer_view, search"/>
 </li>
 <rn:condition_else>
 <li>
 <rn:widget path="navigation/NavigationTab" label_tab="#rn:msg:ANSWERS_HDG#" link="/app/search"
 pages="answers/list, answers/detail, answers/intent, answers/answer_view, search"/>
 </li>
 </rn:condition>
 <li>
 <rn:widget path="navigation/NavigationTab" label_tab="#rn:msg:ASK_QUESTION_HDG#" link="/app/ask"
 pages="ask, ask_confirm"/>
 </li>
</ul>
```

5. Modify the password login by adding this code:

```
sub:login:create_account_fields="Contact.Emails.PRIMARY.Address;Contact.Login;Contact.NewPassword;Contact.FullNa
```

after this line:

```
sub:input_Contact.NewPassword:label_input="#rn:msg:PASSWORD_LBL#"
```

6. Rename the template file to:
okcs_template.php

Add Knowledge Advanced Search to the Home Page

You can add the Knowledge Advanced search (OKCSSearch) functionality so that users can search for knowledge base articles. You can implement the Knowledge Advanced search functionality in the support home (**home.php**) page using either of the following methods:

- Replace the home page in your existing instance with the reference implementation home page.

- Edit it the home page in your existing instance to include code from the reference implementation home page.

Replace the Home Page

Replace the home page with the reference implementation home page.

1. Copy the reference implementation home page from here:
   /cp/core/framework/views/pages/okcs/home.php
2. Place it here:
   /cp/customer/development/views/pages/

Edit the Home Page

Edit the home page to include code from the reference implementation home page.

1. Edit the file:
   cp/customer/development/views/pages/home.php
2. Replace Knowledge Foundation search with Knowledge Advanced search by changing this line:

   ```
   <rn:container source_id="KFSearch">
   ```

   to:

   ```
   <rn:container source_id="OKCSSearch">
   ```

3. Add both the recent searches and the suggested searches widgets by replacing this code:

   ```
   <rn:widget path="searchsource/SourceSearchButton" search_results_url="/app/results"/>
   ```

   with:

   ```
   <rn:widget path="okcs/RecentSearches"/>
   <rn:widget path="okcs/OkcsSuggestions"/>
   <rn:widget path="searchsource/SourceSearchButton" initial_focus="true" search_results_url="/app/
   results"/>
   ```

4. Add the knowledge product and category selector by changing this line:

   ```
   <rn:widget path="navigation/VisualProductCategorySelector" numbered_pagination="true"/>
   ```

   to:

   ```
   <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered_pagination="true"/>
   ```

5. Add both the popular answers and the recent answers widgets by adding this code:

   ```
   <div class="rn_PopularKB">
    <div class="rn_Container">
    <rn:widget path="okcs/AnswerList" type="popular" target="_self" view_type="list"/>
    </div>
   </div>
   <div class="rn_PopularKB rn_RecentKB">
    <div class="rn_Container">
    <rn:widget path="okcs/AnswerList" type="recent" target="_self" view_type="list"/>
    </div>
   </div>
   ```

   after this code:

   ```
   <div class="rn_PageContent rn_Home">
    <div class="rn_Container">
    <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered_pagination="true"/>
    </div>
   ```

Here's an example of the reference implementation home page:

```
<rn:meta title="#rn:msg:SHP_TITLE_HDG#" template="okcs_standard.php" clickstream="home"/>
<div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_HeroCopy">
 <h1>#rn:msg:WERE_HERE_TO_HELP_LBL#</h1>
 </div>
 <div class="rn_SearchControls">
 <h1 class="rn_ScreenReaderOnly">#rn:msg:SEARCH_CMD#</h1>
 <form method="get" action="/app/results">
 <rn:container source_id="OKCSSearch">
 <div class="rn_SearchInput">
 <rn:widget path="searchsource/SourceSearchField" initial_focus="true"
 label_placeholder="#rn:msg:ASK_A_QUESTION_ELLIPSIS_MSG#"/>
 </div>
 <rn:widget path="okcs/RecentSearches"/>
 <rn:widget path="okcs/OkcsSuggestions"/>
 <rn:widget path="searchsource/SourceSearchButton" initial_focus="true" search_results_url="/app/results"/>
 </rn:container>
 </form>
 </div>
 </div>
</div>
<div class="rn_PageContent rn_Home">
 <div class="rn_Container">
 <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered_pagination="true"/>
 </div>
 <div class="rn_PopularKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="popular" target="_self" view_type="list"/>
 </div>
 </div>
 <div class="rn_PopularKB rn_RecentKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="recent" target="_self" view_type="list"/>
 </div>
 </div>
</div>
```

Replace or Edit the Account Overview Page

You can replace or edit the account overview (overview.php) page so that users can view their support questions and content recommendations. Your users can also update their account settings, change their password, manage answer notifications, and manage content type notifications using the account overview page.

You can implement Knowledge Advanced notifications on the account overview page using either of the following methods:

- Replace the overview page in your current instance with the reference implementation overview page.
- Edit the overview page in your current instance to include code from the reference implementation overview page.

Replace the Account Overview Page

Replace the account overview page with the reference implementation overview page.

1. Copy the reference implementation overview page from here:
   /cp/core/framework/views/pages/okcs/account/overview.php
2. Place it here:
   /cp/customer/development/views/pages/account/

Edit the Account Overview Page

Edit the account overview page to include code from the reference implementation overview page.

1. Edit the file:
   /cp/customer/development/views/pages/account/overview.php
2. Add manage content recommendations by adding this code:

```
<div id="rn_LoadingIndicator" class="rn_Browse">
 <rn:widget path="okcs/LoadingIndicator"/>
</div>
 <rn:container source_id="OKCSBrowse">
 <div class="rn_HeaderContainer">
 <h2>a class="rn_Profile" href="/app/account/recommendations/
list#rn:session#">#rn:msg:MY_RECOMMENDATIONS_LBL#</a></h2>
 </div>
 <div id="rn_OkcsManageRecommendations">
 <rn:widget path="okcs/OkcsManageRecommendations"/>
 <a href="/app/account/recommendations/list#rn:session#">#rn:msg:SEE_ALL_RECOMMENDATIONS_LBL#</a>
 </div>
 </rn:container>
```

   after this code:

```
<div class="rn_ContentDetail">
```

3. Remove social discussion notifications by removing this code:

```
<div class="rn_Discussions">
 <rn:container report_id="15156" per_page="4">
 <div class="rn_HeaderContainer">
 <h2><a class="rn_Discussions" href="/app/social/questions/list/author/#rn:profile:socialUserID#/kw/
*#rn:session#">#rn:msg:MY_DISCUSSION_QUESTIONS_LBL#</a></h2>
 </div>
 <rn:widget path="reports/Grid" static_filter="created_by=#rn:profile:socialUserID#"
 label_caption="<span class='rn_ScreenReaderOnly'>#rn:msg:YOUR_RECENTLY_SUBMITTED_DISCUSSIONS_LBL#</
span>"/>
 <a href="/app/social/questions/list/author/#rn:profile:socialUserID#/kw/
*#rn:session#">#rn:msg:SEE_ALL_MY_DISCUSSION_QUESTIONS_LBL#</a>
 </rn:container>
</div>
```

4. Add the content type notifications link in the side rail (`<div class="rn_SideRail">`) section by adding this line:

```
<li>a href="/app/account/notif/
content_type_list#rn:session#">#rn:msg:MANAGE_YOUR_CONTENT_TYPE_NOTIFICATIONS_LBL#</a></li>
```

Here's an example of the reference implementation account overview page:

```
<rn:meta title="#rn:msg:ACCOUNT_OVERVIEW_LBL#" template="okcs_standard.php" login_required="true"
 force_https="true"/>
<div class="rn_Hero">
 <div class="rn_Container">
 <h1>#rn:msg:ACCOUNT_OVERVIEW_LBL#</h1>
```

```
 </div>
</div>
<div class="rn_PageContent rn_AccountOverview rn_Container">
 <div class="rn_ContentDetail">
 <div class="rn_Questions">
 <rn:container report_id="196" per_page="4">
 <div class="rn_HeaderContainer">
 <h2><a class="rn_Questions" href="/app/account/questions/
list#rn:session#">#rn:msg:MY_SUPPORT_QUESTIONS_LBL#</a></h2>
 </div>
 <rn:widget path="reports/Grid" label_caption="<span
 class='rn_ScreenReaderOnly'>#rn:msg:YOUR_RECENTLY_SUBMITTED_QUESTIONS_LBL#</span>"/>
 <a href="/app/account/questions/list#rn:session#">#rn:msg:SEE_ALL_MY_SUPPORT_QUESTIONS_LBL#</a>
 </rn:container>
 </div>
 <div id="rn_LoadingIndicator" class="rn_Browse">
 <rn:widget path="okcs/LoadingIndicator"/>
 </div>
 <rn:container source_id="OKCSBrowse">
 <div class="rn_HeaderContainer">
 <h2><a class="rn_Profile" href="/app/account/recommendations/
list#rn:session#">#rn:msg:MY_RECOMMENDATIONS_LBL#</a></h2>
 </div>
 <div id="rn_OkcsManageRecommendations">
 <rn:widget path="okcs/OkcsManageRecommendations"/>
 <a href="/app/account/recommendations/list#rn:session#">#rn:msg:SEE_ALL_RECOMMENDATIONS_LBL#</a>
 </div>
 </rn:container>
</div>
<div class="rn_SideRail">
 <div class="rn_Well">
 <h3>#rn:msg:LINKS_LBL#</h3>
 <ul>
 <li>a href="/app/account/profile#rn:session#">#rn:msg:UPDATE_YOUR_ACCOUNT_SETTINGS_CMD#/a>/li>
 <rn:condition external_login_used="false">
 <rn:condition config_check="EU_CUST_PASSWD_ENABLED == true">
 <li><a href="/app/account/change_password#rn:session#">#rn:msg:CHANGE_YOUR_PASSWORD_CMD#</a>/li>
 </rn:condition>
 </rn:condition>
 <li><a href="/app/account/notif/list#rn:session#">#rn:msg:MANAGE_YOUR_NOTIFICATIONS_LBL#</a></li>
 <li><a href="/app/account/notif/
content_type_list#rn:session#">#rn:msg:MANAGE_YOUR_CONTENT_TYPE_NOTIFICATIONS_LBL#</a></li>
 <rn:condition is_active_social_user="true">
 <li><a href="/app/public_profile/user/#rn:profile:socialUserID#">#rn:msg:VIEW_YOUR_PUBLIC_PROFILE_LBL#</
a></li>
 </rn:condition>
 </ul>
 </div>
</div>
```

Add the Browse Page

You must implement the Knowledge Advanced browse page so that users can browse articles and filter the articles by content type, product, and category. You implement the browse page by adding the browse page from the reference implementation to your existing instance.

1. Copy the browse page from here:

   /cp/core/framework/views/pages/okcs/browse.php
2. Place it here:

/cp/customer/development/views/pages/

Here's an example of the reference implementation browse page:

```
<rn:meta title="#rn:msg:FIND_ANS_HDG#" template="okcs_standard.php" clickstream="answer_list"/>
<rn:condition config_check="OKCS_ENABLED == true">
 <div class="rn_Container">
 <div id="rn_PageTitle" class="rn_ScreenReaderOnly">
 <h1>#rn:msg:BROWSE1_AB_HDG#</h1>
 </div>
 <rn:container source_id="OKCSBrowse">
 <div id="rn_PageContentChannel" class="rn_Browse rn_PageContentChannel">
 <div class="rn_PageContentChannelInner">
 <rn:widget path="okcs/ContentType"/>
 </div>
 </div>

 <div id="rn_LoadingIndicator" class="rn_Browse">
 <rn:widget path="okcs/LoadingIndicator"/>
 < /div>

 <div id="rn_PageContentArticles" class="rn_AnswerList">
 <rn:widget path="okcs/OkcsRecommendContent"/>
 <div class="rn_ResultPadding">
 <div>
 <div id="rn_OkcsLeftContainer" class="rn_OkcsLeftContainer">
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter_type="products" view_type="explorer"/>
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter_type="categories" view_type="explorer"/>
 </div>
 <div id="rn_OkcsRightContainer" class="rn_OkcsRightContainer">
 <div id="rn_Browse_Loading">/div>
 <div id="rn_OkcsAnswerList">
 <rn:widget path="okcs/AnswerList" target="_self"/>
 <div class="rn_FloatRight">
 <rn:widget path="okcs/OkcsPagination"/>
 </div>
 </div>
 </div>
 </div>
 </div>
 </div>
 </div>
 </rn:container>
 </div>
</rn:condition>
```

Add the Knowledge Advanced Answer Detail Page

You must implement the answer detail (**answer_view.php**) page to open articles with the article details page. Users can see the article content and its details, including document ID, Version, Status, and Published Date using the article details page.

You can implement the answer detail page using either of the following methods:

- Add the reference implementation answer detail page to your current instance.
- Edit the answer detail page in your current instance to include code from the reference implementation answer detail page.

Add the Answer Detail Page

Add the answer details page from reference implementation to your current instance.

1. Copy the reference implementation answer detail from here:

   /cp/core/framework/views/pages/okcs/answers/answer_view.php
2. Place it here:

   /cp/customer/development/views/pages/answers/

Edit the Answer Detail Page

Edit the answer detail page in your current instance to include code from the reference implementation answer detail (**answer_view.php**) page and rename the modified file.

Ensure that the answer detail page does not contain the answer details (`answer_details`) attribute in the meta code `<rn:meta...>`. The presence of this attribute will cause the articles to display incorrectly.

1. Edit the file:

   /cp/customer/development/views/pages/answers/detail.php
2. Replace the entire code with this code:

```
<div class="rn_Container">
 <rn:condition config_check="OKCS_ENABLED == true">
 <div class="rn_ContentDetail">
 <rn:meta title="#rn:php:\RightNow\Libraries\SEO::getDynamicTitle('answer', \RightNow\Utils
\Url::getParameter('a_id'))#" template="okcs_standard.php" clickstream="answer_view"/>
 <div class="rn_PageTitle rn_RecordDetail">
 <rn:widget path="okcs/OkcsProductCategoryBreadcrumb" display_first_item="true"/>
 <div class="rn_OkcsAnswerAction">
 <rn:widget path="okcs/SubscriptionButton"/>
 <rn:widget path="okcs/OkcsRecommendContent"/>
 /div>
 <rn:widget path="okcs/AnswerTitle">
 </div>
 <div class="rn_AnswerView">
 <rn:widget path="okcs/AnswerStatus">
 <div class="rn_SectionTitle"></div>
 <rn:widget path="okcs/AnswerContent">
 </div>
 <div class="rn_DetailTools rn_HideInPrint">
 <div class="rn_Links">
 <rn:widget path="okcs/OkcsEmailAnswerLink" />
 </div>
 </div>
 <rn:widget path="okcs/DocumentRating"/>
 <rn:widget path="okcs/OkcsRelatedAnswers"/>
 </div>
 <aside class="rn_SideRail" role="complementary">
 <rn:widget path="okcs/OkcsRecentlyViewedContent"/>
 </aside>
 </rn:condition>
</div>
```

3. Rename the file:

   answer_view.php

Display User Groups on Answer Pages

You can configure the **Answer Detail** page to display user groups associated with the answer. This feature is not available by default; you must configure it.

To configure this feature, add the following line of code on the answer_view.php page:

```
<rn:widget path="okcs/AnswerStatus" custom_metadata="document_id|version|status|display_date|user_group"
```

Add the Knowledge Advanced Results Page

You must implement the results page so that users can see Knowledge Advanced search results. You implement the search functionality by replacing the results page in your current instance with the reference implementation results page. Users can see a list of articles in search results that match the entered search text.

Users can also enable the interactive spell checker on the results page to either correct misspelled words automatically, or to suggest spelling corrections to the user-entered text.

1. Copy the reference implementation results page from here:

   /cp/core/framework/views/pages/okcs/results.php
2. Place it here:

   /cp/customer/development/views/pages/

Here's an example of the reference implementation results page:

```
<rn:meta title="#rn:msg:FIND_ANS_HDG#" template="okcs_standard.php" clickstream="answer_list"/>
<rn:container source_id="OKCSSearch" document_id_reg_ex="^[\p{L}\p{Nd}_.]*\d{1,10}$"
 doc_id_navigation="true">
 <div id="rn_LoadingIndicator" class="rn_Browse rn_Container">
 <rn:widget path="okcs/LoadingIndicator"/>
 </div>
 <div id="rn_PageTitle" class="rn_Search">
 <div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_SearchControls">
 <h1 class="rn_ScreenReaderOnly">#rn:msg:SEARCH_CMD#</h1>
 <form>
 <div class="rn_SearchInput">
 <rn:widget path="searchsource/SourceSearchField" initial_focus="true"
 label_placeholder="#rn:msg:ASK_A_QUESTION_ELLIPSIS_MSG#" filter_label="Keyword" filter_type="query"/>
 </div>
 <rn:widget path="okcs/RecentSearches"/>
 <rn:widget path="okcs/OkcsSuggestions"/>
 <rn:widget path="searchsource/SourceSearchButton" initial_focus="true" history_source_id="OKCSSearch"/>
 <rn:widget path="okcs/OkcsInteractiveSpellChecker"/>
 <div class="rn_StartOver">
 <a href="/app/home" >#rn:msg:START_OVER_LBL#/a>
 </div>
 </form>
 </div>
 </div>
 </div>
 </div>
```

```
<div id="rn_PageContent" class="rn_Container rn_OkcsResultsContainer">
<div class="rn_ResultList rn_OkcsTable">
<div class="rn_OkcsTableRow">
<div id="rn_OkcsLeftContainer" class="rn_OkcsLeftContainer rn_OkcsTableCell">
<rn:widget path="okcs/Facet"/>
</div>
<div id="rn_OkcsRightContainer" class="rn_OkcsRightContainer rn_OkcsTableCell">
<rn:widget path="okcs/SearchResult" hide_when_no_results="false"/>
<div class="rn_FloatRight">
<rn:widget path="okcs/OkcsPagination"/>
</div>
</div>
</div>
<div class="rn_OkcsTableRow">
<div class="rn_OkcsLeftContainer rn_OkcsTableCell">/div>
<div class="rn_OkcsRightContainer rn_OkcsTableCell">
<rn:widget path="okcs/OkcsRecentlyViewedContent"/>
</div>
</div>
</div>
<aside class="rn_SideRail" role="complementary">
<rn:widget path="utils/ContactUs" channels="question,chat,feedback"/>
</aside>
</div>
</rn:container>
```

Enable Users to Manage Answer Notifications

You must edit the notifications (**list.php**) page so that users can manage their notifications. Users can see notifications they have subscribed to and delete a subscription from the notifications page.

You can edit the notifications page in your current instance to include code from the reference implementation notifications page.

1. Edit the file:

   /cp/customer/development/views/pages/account/notif/list.php
2. Add the loading indicator by adding this code:

   ```
   <div id="rn_LoadingIndicator" class="rn_Browse">
    <rn:widget path="okcs/LoadingIndicator"/>
   </div>
   ```

   before this code:

   ```
   <div class=”rn_PageContent”>
   ```
3. Add the answer notification manager widget by adding this code:

   ```
   <rn:widget path="okcs/OkcsAnswerNotificationManager" view_type="list"/>
   ```

   within this section:

   ```
   <div class=”rn_PageContent”>
   ```

Here’s an example of the reference implementation notifications (**list.php**) page:

```
<rn:meta title="#rn:msg:NOTIFICATIONS_HDG#" template="okcs_standard.php" login_required="true"
 force_https="true"/>
```

```
<div class="rn_Hero">
 <div class="rn_Container">
 <h1>#rn:msg:NOTIFICATIONS_HDG#</h1>
 </div>
</div>
<div id="rn_LoadingIndicator" class="rn_Browse">
 <rn:widget path="okcs/LoadingIndicator"/>
</div>
<div class="rn_PageContent rn_AccountNotifications rn_Container">
 <h2>#rn:msg:ANS_NOTIFICATIONS_LBL#</h2>
 <rn:widget path="okcs/OkcsAnswerNotificationManager" view_type="list"/>
</div>
```

Configure the Related Answer Links

Related Answers are links to topics related to an answer page on Customer Portal. There are two types of related answer links:

- Manual links. These links are manually added to the answer using authoring tools.
- Learned links. These links are generated by machine learning based on users' browsing behavior.

You can edit the Answer Detail Page to show only the manual links, only the learned links, or both types display on an answer page. The default is both types of links display.

To change which link types display, on the reference implementation answer detail (answer_view.php) page, replace:

```
<rn:widget path="okcs/OkcsRelatedAnswers"/>
```

with

```
<rn:widget path="okcs/OkcsRelatedAnswers" display_link_type="manual"/>
```

or

```
<rn:widget path="okcs/OkcsRelatedAnswers" display_link_type="learned"/>
```

or

```
<rn:widget path="okcs/OkcsRelatedAnswers" display_link_type="both"/>
```

Add Knowledge Advanced SmartAssistant

You must edit the ask a question (**ask.php**) page to add the Knowledge Advanced version of SmartAssistant. Users can submit a questions to a support agent using the Ask a Question page. Users can also see knowledge articles as well as community discussions that match the user-entered text before submitting the question.

You can edit the ask a question page in your existing instance by copying code from the reference implementation ask a question page.

1. Edit the file:
   cp/customer/development/views/pages/ask.php
2. Replace standard SmartAssistant with Knowledge Advanced SmartAssistant by changing this code:
   ```
   <rn:condition content_viewed="2" searches_done="1">
   ```

```
 <rn:condition_else/>
 <rn:widget path="input/SmartAssistantDialog"
 label_prompt="#rn:msg:OFFICIAL_SSS_MIGHT_L_IMMEDIATELY_MSG#"/>
</rn:condition>
```

to:

```
<rn:widget path="okcs/OkcsSmartAssistant" view_type="explorer"/>
```

3. Remove this code:

```
[/concept/conbody/ol/licodeblock/i {"fa fa-thumbs-up"}) (i]
```

From this section:

```
<div class="translucent">
 <strong>#rn:msg:TIPS_LBL#:</strong>
 <ul>
 <li>[/concept/conbody/ol/licodeblock/i
{"fa fa-thumbs-up"}) (i]#rn:msg:INCLUDE_AS_MANY_DETAILS_AS_POSSIBLE_LBL#</li>
 </ul>
</div>
<br>
<p>#rn:msg:NEED_A_QUICKER_RESPONSE_LBL# <a href="/app/social/
ask#rn:session#">#rn:msg:ASK_OUR_COMMUNITY_LBL#</a></p>
```

Here's an example of the reference implementation ask a question (**ask.php**) page:

```
<rn:meta title="#rn:msg:ASK_QUESTION_HDG#" template="okcs_standard.php" clickstream="incident_create"/>
<div class="rn_Container">
 <div id="rn_PageTitle" class="rn_AskQuestion">
 <h1>#rn:msg:SUBMIT_QUESTION_OUR_SUPPORT_TEAM_CMD#</h1>
 </div>
</div>
<div id="rn_PageContent" class="rn_AskQuestion rn_Container">
 <div class="rn_Padding">
 <form id="rn_QuestionSubmit" method="post" action="/ci/ajaxRequest/sendForm">
 <div id="rn_ErrorLocation"></div>
 <rn:condition logged_in="false">
 <rn:widget path="input/FormInput" name="Contact.Emails.PRIMARY.Address" required="true"
 initial_focus="true" label_input="#rn:msg:EMAIL_ADDR_LBL#"/>
 <rn:widget path="input/FormInput" name="Incident.Subject" required="true"
 label_input="#rn:msg:SUBJECT_LBL#"/>
 </rn:condition>
 <rn:condition logged_in="true">
 <rn:widget path="input/FormInput" name="Incident.Subject" required="true" initial_focus="true"
 label_input="#rn:msg:SUBJECT_LBL#"/>
 </rn:condition>
 <rn:widget path="input/FormInput" name="Incident.Threads" required="true"
 label_input="#rn:msg:QUESTION_LBL#"/>
 <rn:widget path="input/FileAttachmentUpload"/>
 <rn:widget path="input/ProductCategoryInput" name="Incident.Product"/>
 <rn:widget path="input/ProductCategoryInput" name="Incident.Category" data_type="Category"/>
 <rn:widget path="input/FormSubmit" label_button="#rn:msg:CONTINUE_ELLIPSIS_CMD#" on_success_url="/app/
ask_confirm" error_location="rn_ErrorLocation"/>
 <rn:widget path="okcs/OkcsSmartAssistant" view_type="explorer"/>
 </form>
 </div>
</div>
```

Enable Knowledge Advanced on the Chat Page

You must replace or edit the chat landing (**chat_landing.php**) page to add the Knowledge Advanced chat enhancements to both the chat report page and the chat popup window. Users can see their position in the support queue, estimated and average wait times, and a search field on the chat landing page. The chat window opens when an agent is available.

You must implement the chat enhancements using either of the following methods:

- Replace the chat landing page in your current instance with the reference implementation chat landing page.
- Edit the chat landing page in your current instance to include code from the reference implementation chat landing page.

Replace the Chat Landing Page

Replace the chat landing page with the reference implementation chat landing page.

1. Copy the reference implementation chat landing page from here:
   /cp/core/framework/views/pages/okcs/chat/chat_landing.php
2. Place it here:
   /cp/customer/development/views/pages/chat/

Edit the Chat Landing Page

Edit the chat landing page by copying code from the reference implementation chat landing page.

1. Edit the file:

   /cp/customer/development/views/pages/chat/chat_landing.php
2. Add the chat queue by replacing this line:

   ```
   <rn:widget path="chat/ChatQueueSearch" popup_window="true"/>
   ```

   with:

   ```
   <rn:condition config_check="OKCS_ENABLED == true">
   <rn:widget path="chat/ChatQueueSearch" report_page_url="/app/search" popup_window="true" />
   <rn:condition_else>
   <rn:widget path="chat/ChatQueueSearch" popup_window="true" />
   </rn:condition>
   ```

Implement the Knowledge Advanced Widgets

Your B2C Service instance includes Knowledge widgets. They enable users to access knowledge features, including content type subscription, suggested searches, filter search by product and category, and search result facets.

You can implement widgets and configure them to tailor knowledge for your organization and users. You configure widgets by adding or editing the widget attributes on the pages in your existing instance. You can see the list of knowledge widgets by reading the documentation on Customer Portal administration page: **https://<your_site>/ci/admin/docs/widgets/standard/okcs**

Update Widgets to the Latest Version

Before you implement widgets, update them to the latest version.

1. Go to the Customer Portal administration page:

   https://<your_site>/ci/admin
2. Click on **Widgets**, then **Widget Versions**. The widgets page opens.
3. Click the **Update All** option. The widgets will be updated to the latest version.

Set the Article Display on the Home Page

Users see most popular answers and most recent answers in a table view by default. You change the home page so that users can see articles in a list view. You implement list view using the view type parameter.

1. Edit the file:

   /cp/customer/development/views/pages/home.php
2. Edit the most popular answers code by changing this line:

   ```
   <rn:widget path="okcs/AnswerList" type="popular" target="_self"/>
   ```

   to:

   ```
   <rn:widget path="okcs/AnswerList" type="popular" target="_self" view_type="list"/>
   ```
3. Edit the most recent answers code by changing this line:

   ```
   <rn:widget path="okcs/AnswerList" type="recent" target="_self"/>
   ```

   to:

   ```
   <rn:widget path="okcs/AnswerList" type="recent" target="_self" view_type="list"/>
   ```

Display the Notification Frequency List Box

You display the Notification Frequency list box on the Notifications page so that your users can select how often they want to receive email notifications. Your users can select one of four different frequency options in the list box: Immediately, Once a Day, Once a Week, or Send no emails.

The Notifications Frequency list box is enabled and available to your users by default. However, if you're upgrading to this feature, you must add the path to the Notification Frequency widget.

1. At the Customer Portal administration page, enter **okcs/OkcsSetNotificationFrequency** in the Search box.
2. Click **View** , **OKCS Reference**, and **Production** mode icon.

3. At your Upgrade site, create a new session, open the **list.php** and enter:

```
<rn:widget path= "okcs/OkcsSetNotificationFrequency"
 />
```

4. Navigate to the Notifications page. You may see the following error message:
   "Widget Error: okcs/OkcsSetNotificationFrequency – the widget exists but is not activated. A widget can be activated on the **Widget Management** page."
   Click the Widget Management link and click '**Activate this version'**.
5. At the administration dashboard, click the **View** button, select **OKCS Reference, Promotion** mode. Then select **Deploy**, **Stage**, and then **Promote** to process the changes.
   The Notification Frequency list box now appears on the Notifications page.

See the following example.

```
<rn:meta title="#rn:msg:NOTIFICATION_HDG#" template='okcs_standard.php login_required="true"
 force_https="true">
<div class=rn_Hero">
 <div class="rn_Container">
 <h1>#rn:msg:NOTIFICATION_HDG"</h1>
 </div>
</div>
<div id="rn_LoadingIndicator" class="rn_Browse">
 <rn:widget path="okcs/LoadingIndicator"/>
</div>
<rn:widget path="okcs/OkcsSetNotificationFrequency" />
<div class="rn_PageContent rn=AccountNotification rn_Container">
 <h2>#rn:msg:ANS_NOTIFICATIONS_LBL#</h2>
 <rn:widget path="okcs/OKCSAnswerNotificationManager" view_type="list"/>
</div>
```

Configure Searches to Filter by Product or Category

You can enable users to filters search results using product and category on the home page. Users see only the articles associated with the selected product, category, or both product and category. You implement these filters by including the product category search filter widget. The filters appear only in a tree view.

1. Edit the file:
   cp/customer/development/views/pages/home.php
2. Add the product filter by adding this line:

```
<rn:widget path= "okcs/OkcsProductCategorySearchFilter" filter_type="products"/>
```

3. Add the category filter by adding this line:

```
<rn:widget path= "okcs/OkcsProductCategorySearchFilter" filter_type="categories"/>
```

Here's an example of how to include the product and category filter in the home page:

```
<rn:meta title="#rn:msg:SHP_TITLE_HDG#" template="okcs_standard.php" clickstream="home"/><div
 class="rn_SearchControls">
<div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_HeroCopy">
 <h1>#rn:msg:WERE_HERE_TO_HELP_LBL#</h1>
 </div>
 <div class="rn_SearchControls">
 <h1 class="rn_ScreenReaderOnly">#rn:msg:SEARCH_CMD#</h1>
```

```
 <form method="get" action="/app/results">
 <rn:container source_id="OKCSSearch">
 <div class="rn_SearchInput">
 <rn:widget path="searchsource/SourceSearchField" initial_focus="true"
 label_placeholder="#rn:msg:ASK_A_QUESTION_ELLIPSIS_MSG#"/>
 </div>
 <rn:widget path="okcs/RecentSearches"/>
 <rn:widget path="okcs/OkcsSuggestions"/>
 <rn:widget path="searchsource/SourceSearchButton" initial_focus="true" search_results_url="/app/results"/>
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter_type="products" view_type="explorer"/>
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter_type="categories" view_type="explorer"/>
 </rn:container>
 </form>
 </div>
 </div>
</div>
<div class="rn_PageContent rn_Home">
 <div class="rn_Container">
 <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered_pagination="true"/>
 </div>
 <div class="rn_PopularKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="popular" target="_self" view_type="list"/>
 </div>
 </div>
 <div class="rn_PopularKB rn_RecentKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="recent" target="_self" view_type="list"/>
 </div>
 </div>
</div>
```

Implement Suggested Searches

You can edit the home page so that users can use suggested searches. Users can see a list of articles that match the search term or phrase as they type in the search field. Suggestions with exact keywords match in article titles appear at the top of the results, then suggestions that have partial match. You implement suggested searches to provide a faster way for your users to locate and select relevant articles.

The current implementation contains suggested searches by default on both the home page and the results page. If you are using a previous implementation, you can implement suggested searches using the suggestions widget.

Before you implement suggested searches, you must enable attribute level searching for the master identifier (title field) of each content type in the knowledge base. You can find more information about enabling attribute level searching in the About Schema Attribute Options section in *Administering Knowledge Advanced* .

1. Edit a Customer Portal page, for example:

   /cp/customer/development/views/pages/home.php
2. Add the suggested search widget code as follows:

   ```
   <rn:widget path="okcs/OkcsSuggestions"/>
   ```
3. Add the suggestion count attribute to specify the number of suggested searches you want to display. The default is 7. The maximum value is 100.

   ```
   <rn:widget path="okcs/OkcsSuggestions" suggestion_count="15"/>
   ```

Here's an example of how to include suggested search widget in the home page:

```
<rn:meta title="#rn:msg:SHP_TITLE_H<>DG#" template="okcs_standard.php" clickstream="home"/>
<div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_HeroCopy">
 <h1>#rn:msg:WERE_HERE_TO_HELP_LBL#</h1>
 </div>
 <div class="rn_SearchControls">
 <h1 class="rn_ScreenReaderOnly">#rn:msg:SEARCH_CMD#</h1>
 <form method="get" action="/app/results">
 <rn:container source_id="OKCSSearch">
 <div class="rn_SearchInput">
 <rn:widget path="searchsource/SourceSearchField" initial_focus="true"
 label_placeholder="#rn:msg:ASK_A_QUESTION_ELLIPSIS_MSG#"/>
 </div>
 <rn:widget path="okcs/RecentSearches">
 <rn:widget path="okcs/OkcsSuggestions">
 <rn:widget path="searchsource/SourceSearchButton" initial_focus="true" search_results_url="/app/results"/>
 </form>
 </div>
 </div>
</div>
<div class="rn_PageContent rn_Home">
 <div class="rn_Container">
 <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered_pagination="true"/>
 </div>
 <div class="rn_PopularKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="popular" target="_self" view_type="list"/>
 </div>
 </div>
 <div class="rn_PopularKB rn_RecentKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="recent" target="_self" view_type="list"/>
 </div>
 </div>
</div>
```

For more information on the `OkcsSuggestions` widget, see the Standard Widgets section in *Using B2C Service*.

Configure Search Results Facets

You can configure the display of facets on the search results page. You can use the top facet list attribute in the facet widget to:

- reorder facets
- hide facets
- rename facets

The following list shows the default order in which the facets display on the results page.

- Document Types
- Collections
- Categories

- Products

The following table describes four values with an example each on how to reorder, hide, or rename the facets using the top facet list attribute. You must separate the facets by the pipe (|) character and use the comma (,) character for the facet label description.

Configuration	To display on the results page	Attribute Value
Default	Document Types, Collections, Categories, Products	`<rn:widget path="okcs/Facet" top_facet_list="DOC_TYPES, Document Types|COLLECTIONS, Collections|CMS-CATEGORY-REF, Categories/CMS-PRODUCT, Products"/>`
Reorder	Categories, Products, Collections, Document Types	`<rn:widget path="okcs/Facet" top_facet_list="CMS-CATEGORY-REF|CMS-PRODUCT|COLLECTIONS|DOC_TYPES"/>`
Hide	Document Types, Collections, Categories	`<rn:widget path="okcs/Facet" top_facet_list="DOC_TYPES|COLLECTIONS|CMS-CATEGORY_REF"/>`
Rename	MyCategory, MyDocuments, MyProduct, MyCollections	`<rn:widget path="okcs/Facet" top_facet_list="CMS-CATEGORY-REF, MyCategory|DOC_TYPES, MyDocuments|CMS-PRODUCT, MyProducts|COLLECTIONS, MyCollections"/>`

Enable Users to Use Mutiple Facet Selection

You can implement multiple facet selection so that users can refine search results in the filters widget on the results page. Users can refine their search results by selecting multiple products and categories. They can select only one document type and collection.

> **Note:** Multiple facet selection is not available for *document type* and *content type* facets. Users can select multiple facets for only product and category type facets.

What are the advantages to using multiple facet selection over using a single facet selection?

- The multiple facet selection displays all the product and category facets. The single facet selection displays only the facets that have a search result associated to them
- The multiple facet selection displays top-level facets and the first level children when the page loads. The single facet selection displays only the top level facets.
- The multiple facet selection returns to its original user interface (UI) and the facets go back to their original state when users refresh the page. The single facet selection UI does not change when users refresh the page.

When users select multiple facets, their search results will contain only documents that belong to all of the selections. For example, when user selects product1, product2, category1, category2, document type1, and collection1 from the available facets, the results list displays only documents that belong to all their selection.

Selecting a facet automatically applies all of its children in the hierarchy. Selecting a facet does not apply its parent facet. Each selected facet shows as a separate entry in the filters widget, and users can remove facets. Search results update automatically whenever users change a facet selection.

The current implementation does not include the multiple facet selection functionality on the results page by default. You can implement multiple facet selection by adding the enable multi facet attribute to the facet widget.

1. Edit the file: **/cp/customer/development/views/pages/results.php**
2. Add the enable multi facet attribute to the facet widget as follows:`<rn:widget path="okcs/Facet"enable_multi_select="true"/>`

Enable Users to View the Selected Facets

You can implement facet filter widget so that users can view the facets selected on the results page. Each selected facet shows as a separate entry in the facet filter widget, and users can remove a facet by clicking **X** next to it.

The current implementation includes facet filter at the top of the results page by default. If you upgrade from a previous implementation, you can implement facet filter by adding the facet filter widget to the results page.

1. Edit the file: **/cp/customer/development/views/pages/results.php**
2. Add the facet widget filter by adding this line: `<rn:widget path="okcs/FacetFilter"/>`

Enable Users to Set the Number of Results Per Page

You can configure the results per page drop-down so that users can change the number of display results that appear on each page by adding the results per page attribute to the results page. Specify the options for the number of results as a comma-separated list. The default options are 10, 20, 30, and 50. The results page uses the first number in the list as the default. You can remove the options from the results page by leaving the results per page attribute empty. Search will show ten results on each page by default.

You can follow these steps to configure the results per page drop-down on the results page.

1. Edit the file:

   cp/customer/development/views/pages/results.php
2. Add the results_per_page attribute code as follows:

```
<rn:container source_id="OKCSSearch" document_id_reg_ex="^[\p{L}\p{Nd}_.]*\d{1,10}$"
 doc_id_navigation="true" results_per_page="2,6,12,18,20,25,30">
```

Enable Search by Document ID

You can enable users to bypass the search results list and open an article directly in the article detail page by entering its document ID in the search field. The current implementation contains the search by document ID feature by default. If you upgrade from a previous implementation, you can add search by document ID by modifying the results page.

1. Edit the file:

cp/customer/development/views/pages/results.php

2. Add the `document_id_reg_ex` parameter by changing this line:

```
<rn:container source_id="OKCSSearch"/>
```

to:

```
<rn:container source_id="OKCSSearch" document_id_reg_ex="^[\p{L}\p{Nd}_.]*\d{1,10}$"/>
```

3. Add the `doc_id_navigation` parameter to open an article directly in the article detail page.

```
<rn:container source_id="OKCSSearch" document_id_reg_ex="^\w{1,10}\d{1,10}$" doc_id_navigation="true"/>
```

Change the Style of Intent Answers on the Results Page

You can apply a visual style to answers that are associated to intents. Using a distinct visual style for intent answers helps users identify them as featured answers. Intents are dictionary objects that represent a class of similar questions and answers. Intents provide a way for knowledge managers to feature a single best answer to a set of questions that may vary in wording or detail.

The current implementation contains a default stylesheet to distinguish intent answers from other search results. If you upgrade from a previous implementation to the current implementation, the upgrade process automatically installs the intent style sheet in the default location, **/cp/customer/assets/themes/standard/intent.css**. To use the default intent style, you must update the search result widget to the latest version, which includes the intent style, as shown in this example:

```
<rn:widget path="okcs/SearchResult" apply_style_on_intents="true"/>
```

You can change the default style by editing the attributes in the intent style sheet.

1. Edit the file:
   /cp/customer/assets/themes/standard/intent.css
2. Edit the default style attributes to match your desired style.

Here's an example of an intent style sheet that uses the default style:

```
.rn_IntentResult {
 padding: 0.75em;
 color: #000;
 border-radius: 0.3em;
 border: 0.5em solid #40526b;
}
.rn_IntentResult .rn_ResultIcon:before {
 display: none;
}
.rn_SearchResult .rn_SearchResultContent .rn_IntentResult a {
 font-weight: bold;
}
.rn_IntentResult.rn_ResultElement .rn_Element1 {
 padding-left: unset;
}
```

Configure a Product or Category Landing Page

You can configure the support home page as a product landing page or as a category landing page. Product and category landing pages restrict search and search results to a selected product or category, enabling users to easily find specific information. All searches from the page will automatically use the selected product or category, and answers displayed will be specific to the selected product or category. You can also implement suggested searches to direct users to the most relevant content when they search from a landing page. Users can also navigate to product and categories of interest using breadcrumbs and images.

You configure the home page using the knowledge visual product category selector widget to display either a product landing page or category landing page. The home page is configured as a product landing page by default.

You can add images to display for each product and category. To add product and category images, you must upload an image for each product or category to the following directory:

cp/customer/assets/images/prodcat-images
Images must have the same name as their corresponding product or category. If an image corresponding to a product or category is not found, default image (**default.png**) from the images directory is displayed on the home page.

Modify the home page in your existing instance to configure the product or category landing page.

1. Edit the file:

   cp/customer/development/views/pages/home.php
2. Locate the OkcsVisualProductCategorySelector widget code, for example:

   ```
   <rn:widget path="okcs/OkcsVisualProductCategorySelector" type="product"/>
   ```
3. Edit the `type` attribute to the desired configuration, either product or categories, for example:

   ```
   <rn:widget path="okcs/OkcsVisualProductCategorySelector" type="categories"/>
   ```
4. Add the `maximum_items` attribute to specify the number of products or categories you want to display. The default value is 8. The maximum value is 40.

   ```
   <rn:widget path="okcs/OkcsVisualProductCategorySelector" maximum_items="12"/>
   ```

Enable Suggested Searches on a Product or Category Landing Page

You can add suggested searches to product and category landing pages to configure the search behavior to meet your organization's needs. This feature already exists on the home page. Suggested searches on landing pages make it easier for users to find the knowledge base content that is most relevant to the purpose of the page.

Once you configure suggested searches, Customer Portal users see suggested searches as a dropdown list when they use the search field on a product landing page. This feature already exists on the home page. The landing page suggested searches will reflect the product and category of the landing page. This provides a better search experience.

You can set options for search indexing and which products and categories to include for suggested articles.

You can set options for search indexing and which products and categories to include for suggested articles.

Set the `product_category parameter` to enable search to filter suggestions by product and/or category.

- **Note:** You must have `Enable_Product_Category_PreFilter` enabled to use this parameter.

Configure the Article Attributes

You can configure article attributes to display when users view an article.You can configure the article attributes so that users can see the article details along with its content. You configure the article attributes that you want to display by including the custom metadata parameter in the answer view page and the ask a question page. By default, your users can see article details, including Document ID, Version, Status, and Display Date.

Attribute	Description
document_id	The article's document ID.
version	The article's version.
status	The article's status. Displays whether the article is published or in draft state.
display_date	The publish date of a published article or the last modified date of a draft article.
creator	The user who created the article.
aggregate_rating	The aggregate rating of article.
owner	The article's owner.
answer_id	The article's answer ID.
last_modified	The date that the article was last modified.
last_modifier	The user who last updated the article.
user_group	The user group that the article is visible to.

Configure Article Attributes on the Answer Page

You can configure article attributes on the answer view page using the custom metadata attribute.

1. Edit the file:

cp/customer/development/views/pages/answer/answer_view.php

2. Add the custom_metadata parameter code as follows:

```
<rn:widget path="okcs/AnswerStatus" custom_metadata="attribute | attribute | attribute"/>
```

3. Specify the attributes that you want to display as a list of items separated by the pipe character (|), for example:

```
<rn:widget path="okcs/AnswerStatus" custom_metadata="document_id | version | status | display_date |
 user_group"/>
```

Configure Article Attributes on the Ask a Question Page

You can configure article attributes on the Ask a Question page using the custom metadata attribute.

1. Edit the file:

   cp/customer/development/views/pages/ask.php

2. Add the custom_metadata parameter code as follows:

```
<rn:widget path="okcs/OkcsSmartAssistant" custom_metadata="attribute | attribute | attribute"/>
```

3. Specify the attributes that you want to display as a list of items separated by the pipe character (|), for example:

```
<rn:widget path="Okcs/SmartAssistant" custom_metadata="document_id | version | status | display_date|
user_group"/>
```

Enable Users to View a Translated Version of an Article

You can implement the language drop-down so that users can see translated versions of an article from the article detail page. You implement the language drop-down by adding the translated answer selector widget to the article detail page.

Users select a locale from the language drop-down to see the article and its content in the selected locale. The language drop-down displays locales for only the published articles. In addition, a user can see translated versions of an article based on the usergroup access at the article level.

1. Edit the file:

   cp/customer/development/views/pages/answer/answer_view.php

2. Add the language drop-down by adding this line:

```
<rn:widget path="okcs/OkcsTranslatedAnswerSelector"/>
```

   within this section:

```
<div class="rn_OkcsAnswerAction">
 <rn:widget path="okcs/SubscriptionButton"/>
 <rn:widget path="okcs/OkcsRecommendContent"/>
</div>
```

3. Specify the label attribute to change label for the drop-down as follows:

```
<rn:widget path="okcs/OkcsTranslatedAnswerSelector" label_drop_down="<label_name>"/>
```

Here's an example on how to include the language drop-down in the answer detail (**answer_view.php**) page:

```
<div class="rn_Container">
```

```
<rn:condition config_check="OKCS_ENABLED == true">
<div class="rn_ContentDetail">
<rn:meta title="#rn:php:\RightNow\Libraries\SEO::getDynamicTitle('answer',
\RightNow\Utils\ Url::getParameter('a_id'))#" template="okcs_standard.php"
clickstream="answer_view"/>
<div class="rn_PageTitle rn_RecordDetail">
<rn:widget path="okcs/OkcsProductCategoryBreadcrumb" display_first_item="true"/>
<div class="rn_OkcsAnswerAction">
<rn:widget path="okcs/SubscriptionButton"/>
<rn:widget path="okcs/OkcsRecommendContent"/>
<rn:widget path="okcs/OkcsTranslatedAnswerSelector"/>
</div>
<rn:widget path="okcs/AnswerTitle">
</div>
<div class="rn_AnswerView">
<rn:widget path="okcs/AnswerStatus">
<div class="rn_SectionTitle"></div>
<rn:widget path="okcs/AnswerContent">
</div>
<div class="rn_DetailTools rn_HideInPrint">
<div class="rn_Links">
<rn:widget path="okcs/OkcsEmailAnswerLink"/>
</div>
</div>
<rn:widget path="okcs/DocumentRating"/>
<rn:widget path="okcs/OkcsRelatedAnswers"/>
</div>
<aside class="rn_SideRail" role="complementary">
<rn:widget path="okcs/OkcsRecentlyViewedContent"/>
</aside>
</rn:condition>
</div>
```

Enable Users to Display Answers for All Content Types

You can edit the browse page to add the All option so that users can see articles for all content types. Users can see articles for all the content types or filter articles by a content type. You add the All option by including the display all label attribute in the content type widget.

1. Edit the file:

   /cp/customer/development/views/pages/browse.php
2. Add the All option by changing this line:

   ```
   <rn:widget path="okcs/ContentType"/>
   ```

   to:

   ```
   <rn:widget path="okcs/ContentType" display_all_content="true" display_all_label="<new_label_name>" />
   ```

Implement Content Type Subscriptions

Article subscription by content type is implemented by default on the account overview page. If you are using a previous implementation, you can implement content type subscription to enable users to subscribe to articles by content type.

When users subscribe to a content type, they get an email when there are new or significantly updated articles of that type.

Users can limit their subscription to articles about specific products and categories in the content types that they subscribe to.

You implement content type subscription by adding a reference to the content type subscription page.

1. Copy the `content_type_list.php` file from **/cp/core/framework/views/pages/okcs/account/notif** to **/cp/customer/development/views/pages/account/notif**.
2. Edit a Customer Portal page, for example, the account overview page:

   /cp/customer/development/views/pages/account/overview.php
3. Add the code snippet to refer to the content type subscription page, as follows:

```
<a href="/app/account/notif/content_type_list#rn:session#">#rn:astr:Manage your content type answer
 notifications#</a>
```

Here's an example on how to include a reference to the content type subscription page in the account overview page:

```
<rn:meta title="#rn:msg:ACCOUNT_OVERVIEW_LBL#" template="okcs_standard.php" login_required="true"
 force_https="true"/>

<div class="rn_Hero">
 <div class="rn_Container">
 <h1>#rn:msg:ACCOUNT_OVERVIEW_LBL#</h1>
 </div>
</div>

<div class="rn_PageContent rn_AccountOverview rn_Container">
 <div class="rn_ContentDetail">
 <div class="rn_Questions">
 <rn:container report_id="196" per_page="4">
 <div class="rn_HeaderContainer">
 <h2><a class="rn_Questions" href="/app/account/questions/
list#rn:session#">#rn:msg:MY_SUPPORT_QUESTIONS_LBL#/<a></h2>
 </div>
 <rn:widget path="reports/Grid" label_caption="<span
 class='rn_ScreenReaderOnly'>#rn:msg:YOUR_RECENTLY_SUBMITTED_QUESTIONS_LBL#</span>"/>
 <a href="/app/account/questions/list#rn:session#">#rn:msg:SEE_ALL_MY_SUPPORT_QUESTIONS_LBL#</a>
 </rn:container>
 </div>

 <div id="rn_LoadingIndicator" class="rn_Browse">
 <rn:widget path="okcs/LoadingIndicator"/>
 </div>
 <rn:container source_id="OKCSBrowse">
 <div class="rn_HeaderContainer">
 <h2><a class="rn_Profile" href="/app/account/recommendations/
list#rn:session#">#rn:msg:MY_RECOMMENDATIONS_LBL#</a></h2>
 </div>
 <div id="rn_OkcsManageRecommendations">
 < rn:widget path="okcs/OkcsManageRecommendations"/>
 <a href="/app/account/recommendations/list#rn:session#">#rn:msg:SEE_ALL_RECOMMENDATIONS_LBL#</<a>
 </div>
 </rn:container>
</div>

<div class="rn_SideRail">
 <div class="rn_Well">
 <h3>#rn:msg:LINKS_LBL#</h3>
 <ul>
 <li><a href="/app/account/profile#rn:session#">#rn:msg:UPDATE_YOUR_ACCOUNT_SETTINGS_CMD#</a></li>
```

```
 <rn:condition external_login_used="false">
 <rn:condition config_check="EU_CUST_PASSWD_ENABLED == true">
 <li><a href="/app/account/change_password#rn:session#">#rn:msg:CHANGE_YOUR_PASSWORD_CMD#</a></li>
 </rn:condition>
 </rn:condition>
 <li><a href="/app/account/notif/content_type_list#rn:session#">#rn:astr:Manage your content type answer
 notifications#</a></li>
 <rn:condition is_active_social_user="true">
 <li><a href="/app/public_profile/user/#rn:profile:socialUserID#">#rn:msg:VIEW_YOUR_PUBLIC_PROFILE_LBL#</
a></li>
 </rn:condition>
 </ul>
 </div>
</div>
```

Enable Content Recommendation

This function allows users to recommend new content or changes to an existing content in Knowledge Base. The recommendation appears under Tasks & Recommendations in the Repository Content Type Properties page in Authoring. The administrator can accept or reject the recommendation. Users can see the status of their recommendations in Account Overview > My Recommendations.

Before you implement content recommendation, you must enable the content recommendation option for the content type in the knowledge base. You can find more information about enabling content recommendation in the *Add Content Type General Properties* in *Administering Knowledge Advanced*.

The following code adds the Recommend Content button and text fields to the **pages/answer/answer_view.php** and **pages/browse.php** pages.

```
<rn:widget path="okcs/OkcsRecommendContent"/>
```

Implement Favorites

You can implement favorites so that users can bookmark frequently-used articles. Favorites provides users with quick and easy access to the articles they use most often. Users can access their favorites on article detail page and account overview page.

Users must log into Customer Portal to use favorites. They can add articles to their favorites by clicking Add Favorite on the article detail page, and remove articles from their favorites by clicking Remove Favorite. They can also view and manage their favorites on the account overview page.

The current implementation includes favorites on both the article detail page and the account overview page by default. If you upgrade from a previous implementation, you can implement favorites by adding the favorites button widget to the article detail page, adding the favorites list widget to the account overview page, and copying the favorites list reference page into your site.

1. Edit the file: **/cp/customer/development/views/pages/answer/answer_view.php**
2. Add the favorite button by adding this line:

   ```
   rn:widget path="okcs/FavoritesButton"/>
   ```

   after this line:

```
rn:widget path="okcs/OkcsRecommendContent"/>
```

3. Add the favorites list reference (**list.php**) page by copying the favorites folder from **/cp/core/framework/views/pages/okcs/account** to **/cp/customer/development/views/pages/account**
4. Edit the file: **/cp/customer/development/views/pages/account/overview.php**
5. Add the favorites list section by adding the following code:

```
<div class="rn_HeaderContainer">
 <h2><a class="rn_Profile" href="/app/account/favorites/list#rn:session#">#rn:msg:MY_FAVORITES_LBL#</
a></h2>
</div>
<div id="rn_OkcsFavoritesList">
 <rn:widget path="okcs/OkcsFavoritesList" view_type="table" display_fields="title|documentId"/>
 <a href="/app/account/favorites/list#rn:session#">#rn:astr: See all favorites#</a>
</div>
```

within this section:

```
<rn:container source_id="OKCSBrowse">
</rn:container>
```

1.

Suppress SmartAssistant Filters

SmartAssistant uses the user-provided product and category as search filters to restrict results by default. You can override the default filtering to use only product, category, or neither to filter search results.

SmartAssistant uses the user-provided product and category as search filters to restrict results. By default, it uses both product and category. When using SmartAssistant to search, you can override the default filtering to use only product, category, or neither.

A hook in the file `hooks.php`, which contains hook definitions, calls a method defined in `Sample.php`. To apply a specific choice of filter, or to ignore both filters, modify the files `hooks.php` and `Sample.php`.

1. Edit the file:

   cp/customer/development/config/hooks.php
2. Add the following code at the end of the file:

```
$rnHooks['pre_retrieve_smart_assistant_answers'][] = array( 'class' => 'Sample', // Name of the custom
 model 'function' => 'retrieveSmartAssistantRequest', 'filepath' => '' );
```

3. Edit the file:

   cp/customer/development/models/custom/Sample.php
4. Add the following code at the end of the file:

```
<?php
namespace Custom\Models;
require_once CPCORE . 'Models/Okcs.php';

class Sample extends \RightNow\Models\Okcs
{
function __construct()
```

```
{
parent::__construct();
}
public function retrieveSmartAssistantRequest(&$hookData) {
// Second argument determines which SA results to show. Possible values are 'Product', 'Category',
 'ProductAndCategory', 'None'.
parent::retrieveSmartAssistantRequest(&$hookData, 'ProductAndCategory');
}
}
```

Enable Users to Open an Article Attachment

Users can open file attachments of an article in a separate browse window with the answer page. You implement the answer page by adding **answer.php** from the reference implementation to your current instance.

1. Copy the **answer.php** file from here:

   /cp/core/framework/views/admin/okcs/
2. Place it here:

   /cp/customer/development/views/admin/

Enable Agents to Open an Article from Agent Desktop

You can enable the support agents to open articles that users have viewed before submitting their question using the Ask a Question page. You enable agents to open articles in Agent Desktop by adding the answer preview page (**okcs_answer_full_preview.php**) from the reference implementation to your current instance.

1. Copy the **okcs_answer_full_preview.php** file from here:

   /cp/core.framework/views/admin/okcs/
2. Place it here:

   /cp/customer/development/views/admin/

Add Knowledge to Website Pages

You can use the knowledge syndication widget to add knowledge to pages on your organization's website in addition to the customer portal. This enables users to search for and use knowledge from the most relevant places on your web page. You can set up the knowledge widget to display articles by content type, product, and category, popular and recent articles, and external articles.

You add the knowledge widget by configuring the widget, then pasting the JavaScript code that the widget generates into your organization's web page.

Configure Knowledge Widget

Use the **Configure Widget** section of the **widget configuration** page to configure the knowledge widget.

1. Go to the your site's administration page:

   https://<your_site>/ci/admin
2. Click **Widgets**, then **Syndicated Widgets**.
3. Click **OkcsKnowledgeSyndication**.
4. Configure the `okcsKnowledgeSyndication` widget attributes using the **Configure Widget** section.

   You can see the list of the widget attributes and their descriptions on your site's administration page: **https://<your_site>/ci/tags/syndicated_widgets/standard/OkcsKnowledgeSyndication**.
5. Click **Apply**.
6. Preview your changes in the **Preview this Configuration** section.

Add the Widget to a Page

Use the **Copy Script code to your page** section of the **widget configuration** page to copy and paste the syndication widget code into your web page.

1. Open the source code for the web page that you want to add the widget to.
2. Click the **Select Text** button to select the code that defines your site URL, then press Ctrl+c to copy the code.

   Here's an example of the code that includes your site URL:

   ```
   <script type="text/javascript" scr="https://<your_site>/euf/rightnow/RightNow.Client.js"></script>
   ```
3. Paste the code into the source code of your web page just before the closing `</body>` tag.
4. Click the **Select Text** button to select the code that defines the widget and its attributes, then press Ctrl+c to copy the code.

   Here's an example of the JavaScript code that includes the widget configuration:

   ```
   <script type="text/javascript">
    RightNow.Client.Controller.addComponent(
    {
    div_id: "myDiv",
    enable_recent: true,
    presist_contenttype: false,
    persist_prodcat: false,
    q: "oracle",
    instance_id: "okcs_0",
    module: "OkcsKnowledgeSyndication",
    type: 10
    }
    "https://<your_site>/ci/ws/get"
    );
   </script>
   ```
5. Paste the copied code just below the code you copied in the previous step.
6. Click the **Select Text** button to select the `<div>` tag code, then press Ctrl+c to copy the code. This code calls the JavaScript code of the widget generated in the previous step.

   Here's an example of the `<div>` tag code:

   ```
   <div id="myDiv"></div>
   ```
7. Place the `<div>` tag within the `</body>` tag where you want the widget to appear on the web page.

Community Discussions

Community Self Service in B2C Service provides a platform for community and peer-to-peer knowledge sharing. It provides self-service users with an additional channel that community members can use to ask questions, provide answers, and discuss issues.

You can add community discussions to knowledge in your current instance to enable self-service users to get answers from discussions along side knowledge base answers. Community discussions are organized using the same product hierarchy as knowledge base articles.

You can include community discussions on the home page, the results page, and in the Knowledge Advanced version of SmartAssistant. When users search for answers on these pages, their answers will include both knowledge base articles and community discussions.

You can find more information about Community Self Service in *Using B2C Service*.

Add Community Discussions to the Home Page

Edit the home page so that users can to see community discussions in addition to most popular answers and most recent answers.

1. Edit the file: **/cp/customer/development/views/pages/home.php**
2. Add community discussions by adding this code:

```
<div class="rn_PopularSocial">
<div class="rn_Container">
<h2>#rn:msg:RECENT_COMMUNITY_DISCUSSIONS_LBL#</h2>
<rn:widget path="discussion/RecentlyAnsweredQuestions" show_excerpt="true" maximum_questions="5"/>
<span class="rn_DiscussionLink">
<a href="/app/social/questions/list/kw/*#rn:session#">#rn:msg:SHOW_MORE_COMMUNITY_DISCUSSIONS_LBL#</a>
</span>
</div>
</div>
```

Add Community Discussions to the Results Page

Edit the results page so that users can see community discussions in addition to the articles in search results.

1. Edit the file: **/cp/customer/development/views/pages/results.php**
2. Add community discussions by adding this code:

```
<div class="rn_Container">
<div class="rn_QuestionResults">
<rn:container source_id="SocialSearch" per_page="5">
<rn:widget path="searchsource/SourceResultDetails"/>
<rn:widget path="searchsource/SocialResultListing"
 label_heading="#rn:msg:RESULTS_FROM_THE_COMMUNITY_LBL#" more_link_url="/app/social/questions/list"/>
</rn:container>
</div>
</div>
```

Add Recently Viewed Articles to the Social Page

Users can see the discussions about a selected community questions on the social detail page. You can add recently viewed knowledge base articles to the social detail page so that they will see recently viewed articles in addition to community discussions. You add them by including the recently viewed answers widget in the social detail page.

Users can see five recently viewed articles by default. You can set the limit to a value to five or fewer using the widget's content count attribute.

1. Edit the file:

   cp/customer/development/views/pages/social/questions/detail.php
2. Add the recently viewed articles widget code by adding this line:

   ```
   <rn:widget path="okcs/OkcsRecentlyViewedContent"/>
   ```

   within this section:

   ```
   <div class="rn_SideRail">
   ```

Add Related Answers to the Social Page

Users can see the discussions about a selected community question on the social detail page. You can add related knowledge base answers so that they will also see relevant articles in addition to community discussions. You add them by including the related answers widget in the social detail page.

Users can see up to ten related articles by default. You can set the limit number of articles to ten or fewer using the widget's limit attribute.

1. Edit the file: **cp/customer/development/views/pages/social/questions/detail.php**
2. Add this line:

   ```
   <rn:widget path="okcs/RelatedKnowledgeAdvanceAnswers" limit="5"/>
   ```

   to this section:

   ```
   <div class="rn_SideRail">
   ```

Add Community Discussions to SmartAssistant

The Knowledge Advanced version of SmartAssistant includes community discussions as well as knowledge base answers. When users submit a question, Knowledge Advanced search returns answers and social search returns discussions that match the search terms that were entered.

Knowledge Advanced uses its own version of SmartAssistant. You can add community discussions to the Knowledge Advanced SmartAssistant so that users can see both community content and knowledge base articles in search

results. You update SmartAssistant by editing the KFAPI_SSS_ENABLED configuration setting. The default value of the configuration setting is No.

1. Go to **Site Configuration**, **Configuration Settings**.
2. Set the value of the configuration setting KFAPI_SSS_ENABLED to `Yes`.

Implementing Knowledge Advanced on Mobile Customer Portal

You enable Knowledge Advanced in an existing Customer Portal mobile implementation by replacing or modifying the existing Customer Portal mobile pages. You can replace the entire set of Customer Portal files with the Knowledge Advanced mobile reference implementation. However, the new Knowledge Advanced mobile files will overwrite any customized files in your current implementation. You can preserve your customized files by using the Knowledge Advanced mobile files as examples on how to modify the files instead of replacing them.

Add Knowledge Advanced Search to the Mobile Home Page

You can add the mobile search (OKCSSearch) functionality so that users can search for knowledge base articles from the home page. You implement the search functionality by replacing the mobile home page in your existing instance with the reference implementation mobile home page.

1. Copy the reference implementation mobile home page from here:

   /cp/core/framework/views/pages/okcs/mobile/home.php
2. Place it here:

   /cp/customer/development/views/pages/mobile

Here's an example of the reference implementation mobile home page:

```
<rn:meta title="#rn:msg:SHP_TITLE_HDG#" template="okcs_mobile.php" clickstream="home"/>
<div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_HeroCopy">
 <h1>#rn:msg:WERE_HERE_TO_HELP_LBL#</h1>
 </div>
 <div class="rn_SearchControls">
 <h1 class="rn_ScreenReaderOnly">#rn:msg:SEARCH_CMD#</h1>
 <form method="get" action="/app/results">
 <rn:container source_id="OKCSSearch">
 <div class="rn_SearchInput">
 <rn:widget path="searchsource/SourceSearchField" initial_focus="true"
 label_placeholder="#rn:msg:ASK_A_QUESTION_ELLIPSIS_MSG#"/>
 </div>
 <rn:widget path="okcs/RecentSearches"/>
 <rn:widget path="okcs/OkcsSuggestions"/>
 <rn:widget path="searchsource/SourceSearchButton" initial_focus="true" search_results_url="/app/results"/>
 </rn:container>
 </form>
 </div>
 </div>
</div>
<div class="rn_PageContent rn_Home">
 <div class="rn_Container">
 <rn:widget path="okcs/OkcsVisualProductCategorySelector" numbered_pagination="true"/>
```

```
 </div>
 <div class="rn_PopularKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="popular" target="_self" view_type="list"/>
 </div>
 </div>
 <div class="rn_PopularKB rn_RecentKB">
 <div class="rn_Container">
 <rn:widget path="okcs/AnswerList" type="recent" target="_self" view_type="list"/>
 </div>
 </div>
</div>
```

Add Mobile Knowledge Advanced Browse

You must implement the mobile browse (OKCSBrowse) functionality so that users can browse articles and filter the articles by content type, product, and category. You implement the browse functionality by adding the browse page from your reference implementation to your existing instance.

1. Copy the reference implementation mobile browse page from here:
   /cp/core/framework/views/pages/okcs/mobile/browse.php
2. Place it here:
   /cp/customer/development/views/pages/mobile/

Here's an example of the reference implementation mobile browse (**browse.php**) page:

```
<rn:meta title="#rn:msg:BROWSE1_AB_HDG#" template="okcs_mobile.php" clickstream="answer_list"/>
<rn:condition config_check="OKCS_ENABLED == true">
<div id="rn_PageTitle" class="rn_Home">
 <div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_OkcsMobileBrowseHeader">
 <h1 class="rn_ScreenReaderOnly">rn:page_title/></h1>
 <rn:container source_id="OKCSBrowse">
 <h2 id="rn_AccordTriggerContentType" class="rn_Collapsed"><a href="javascript:void(0);"
 role="button">#rn:msg:CONTENT_TYPE_LBL#span class="rn_ButtonOff">/span></a></h2>
 <h2 id="rn_AccordTriggerProduct" class="rn_Collapsed"><a href="javascript:void(0);"
 role="button">#rn:msg:PRODUCT_LBL#span class="rn_ButtonOff">/span></a></h2>
 <h2 id="rn_AccordTriggerCategory" class="rn_Collapsed"><a href="javascript:void(0);"
 role="button">#rn:msg:CATEGORY_LBL#span class="rn_ButtonOff">/span></a></h2>
 <div class="rn_ClearBoth">
 <div id="rn_ContainerContentType" class="rn_Hidden">
 <rn:widget path="okcs/ContentType" list_display="vertical" toggle_selection="true"
 toggle="rn_AccordTriggerContentType" item_to_toggle="rn_ContainerContentType"/>
 </div>
 <div id="rn_ContainerProduct" class="rn_Hidden">
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter_type="products" toggle_selection="true"
 toggle="rn_AccordTriggerProduct" item_to_toggle="rn_ContainerProduct" view_type="explorer"/>
 </div>
 <div id="rn_ContainerCategory" class="rn_Hidden">
 <rn:widget path="okcs/OkcsProductCategorySearchFilter" filter_type="categories" toggle_selection="true"
 toggle="rn_AccordTriggerCategory" item_to_toggle="rn_ContainerCategory" view_type="explorer"/>
 </div>
 </div>
 </div>
 </div>
 </div>
</div>
<div class="rn_Container">
<section id="rn_PageContent" class="rn_Home">
```

```
 <div id="rn_LoadingIndicator" class="rn_Browse">
 <rn:widget path="okcs/LoadingIndicator"/>
 </div>
 <div id="rn_PageContentArticles">
 <div class="rn_ResultPadding">
 <div id="rn_Browse_Loading">/div>
 <div id="rn_OkcsAnswerList">
 <div class="rn_Report">
 <rn:widget path="okcs/OkcsRecommendContent"/>
 <rn:widget path="okcs/AnswerList" view_type="list" show_headers="false" per_page="5" target="_self"/>
 </div>
 <div class="rn_FloatRight">
 <rn:widget path="okcs/OkcsPagination"/>
 </div>
 </div>
 </div>
 </div>
 </rn:container>
</section>
</div>
</rn:condition>
```

Add the Mobile Knowledge Advanced Answer Detail Page

You must implement the mobile answer detail (**answer_view.php**) page to enable users to open articles with the article details page. Users can see the article content and its details, including document ID, Version, Status, and Published Date using the article details page.

You implement the mobile answer detail page by adding **answer_view.php** from reference implementation to your existing instance.

1. Copy the reference implementation mobile answer detail from here:

   /cp/core/framework/views/pages/okcs/mobile/answers/answer_view.php
2. Place it here:

   /cp/customer/development/views/pages/mobile/answers/

The following is an example of the reference implementation mobile answer detail (**answer_view.php**) page:

```
<div class="rn_Container">
 <rn:condition config_check="OKCS_ENABLED == true">
 <rn:widget path="okcs/OkcsProductCategoryBreadcrumb" display_first_item="false"/>
 <rn:meta title="#rn:php:\RightNow\Libraries\SEO::getDynamicTitle('answer', \RightNow\Utils
\Url::getParameter('a_id'))#" template="okcs_mobile.php" clickstream="answer_view"/>
 <rn:widget path="okcs/OkcsRecommendContent"/>
 <rn:widget path="okcs/SubscriptionButton"/>
 <section id="rn_PageContent" class="rn_AnswerDetail">
 <div id="rn_AnswerText">
 <rn:widget path="okcs/AnswerTitle">
 <rn:widget path="okcs/AnswerStatus">
 <div class="rn_SectionTitle"></div>
 <rn:widget path="okcs/AnswerContent">
 </div>
 <div class="rn_DetailTools rn_HideInPrint">
 <div class="rn_Links">
 <rn:widget path="okcs/OkcsEmailAnswerLink" />
 </div>
 </div>
 <rn:widget path="okcs/DocumentRating"/>
```

```
 <rn:widget path="okcs/OkcsRelatedAnswers"/>
 </section>
 </rn:condition>
</div>
```

Add the Mobile Knowledge Advanced Results Page

You must implement the mobile Knowledge Advanced results page so that users can see knowledge articles in search results. You implement the search functionality by replacing the mobile results page in your current instance with the reference implementation mobile results page.

Users can also use the interactive spell checker on the results page to either correct misspelled words automatically, or to suggest spelling corrections to the user-entered text.

1. Copy the reference implementation mobile results page from here:

   /cp/core/framework/views/pages/okcs/mobile/results.php
2. Place it here:

   /cp/customer/development/views/pages/mobile/

Here's an example of the reference implementation mobile results (**results.php**) page:

```
<rn:meta title="#rn:msg:FIND_ANS_HDG#" template="okcs_mobile.php" clickstream="answer_list"/>
<rn:container source_id="OKCSSearch" document_id_reg_ex="^\w{1,10}\d{1,10}$">
<div id="rn_PageTitle" class="rn_Search">
 <div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_SearchControls">
 <h1 class="rn_ScreenReaderOnly">#rn:msg:SEARCH_CMD#</h1>
 form method="get" action="/app/results">
 <rn:container source_id="OKCSSearch">
 <div class="rn_SearchInput">
 <rn:widget path="searchsource/SourceSearchField" initial_focus="true"
 label_placeholder="#rn:msg:ASK_A_QUESTION_ELLIPSIS_MSG#"/>
 </div>
 <rn:widget path="okcs/RecentSearches" display_tooltip="true"/>
 <rn:widget path="okcs/OkcsSuggestions"/>
 <rn:widget path="searchsource/SourceSearchButton" initial_focus="true" history_source_id="OKCSSearch"/>
 <rn:widget path="okcs/OkcsInteractiveSpellChecker"/>
 </rn:container>
 <div class="rn_StartOver">
 <a href="/app/home" >#rn:msg:START_OVER_LBL#</a>
 </div>
 </form>
 </div>
 </div>
 </div>
</div>
</rn:container>
<div class="rn_Container">
 <section id="rn_PageContent" class="rn_Container">
 <rn:container source_id="OKCSSearch" truncate_size="200">
 <div class="rn_Module">
 <rn:widget path="okcs/Facet" toggle_title="true"/>
 </div>
 <div id="rn_OkcsRightContainer" class="rn_Padding">
 <rn:widget path="okcs/SearchResult" hide_when_no_results="false"/>
 <div class="rn_FloatRight rn_Padding">
 <rn:widget path="okcs/OkcsPagination"/>
 </div>
```

```
 </div>
 </rn:container>
 </section>
</div>
```

Add Knowledge Advanced SmartAssistant

You must add the mobile ask a question (**ask.php**) page to implement the Knowledge Advanced version of SmartAssistant. Users can submit a question to a support agent from the Ask a Question page. Users can also see knowledge articles as well as community discussions that match the user-entered text before submitting the question.

You implement mobile SmartAssistant by replacing the mobile ask a question page in your current instance with the reference implementation mobile ask a question page.

1. Copy the reference implementation mobile ask a question page from here:

   /cp/core/framework/views/pages/okcs/mobile/ask.ph
2. Place it here:

   /cp/customer/development/views/pages/mobile/

Here's an example of the reference implementation mobile ask a question (**ask.php**) page:

```
<rn:meta title="#rn:msg:ASK_QUESTION_HDG#" template="okcs_mobile.php" clickstream="incident_create"/>
<div class="rn_Hero">
 <div class="rn_HeroInner">
 <div class="rn_HeroCopy">
 <h1>#rn:msg:SUBMIT_QUESTION_OUR_SUPPORT_TEAM_CMD#</h1>
 <p>#rn:msg:OUR_DEDICATED_RESPOND_WITHIN_48_HOURS_MSG#</p>
 </div>
 </div>
</div>

<div class="rn_PageContent rn_AskQuestion rn_Container">
 <form id="rn_QuestionSubmit" method="post" action="/ci/ajaxRequest/sendForm">
 <div id="rn_ErrorLocation">/div>
 <rn:condition logged_in="false">
 <rn:widget path="input/FormInput" name="Contact.Emails.PRIMARY.Address" required="true"
 initial_focus="true" label_input="#rn:msg:EMAIL_ADDR_LBL#"/>
 <rn:widget path="input/FormInput" name="Incident.Subject" required="true"
 label_input="#rn:msg:SUBJECT_LBL#"/>
 </rn:condition>
 <rn:condition logged_in="true">
 <rn:widget path="input/FormInput" name="Incident.Subject" required="true" initial_focus="true"
 label_input="#rn:msg:SUBJECT_LBL#"/>
 </rn:condition>
 <rn:widget path="input/FormInput" name="Incident.Threads" required="true"
 label_input="#rn:msg:QUESTION_LBL#"/>
 <rn:widget path="input/FileAttachmentUpload"/>
 <rn:widget path="input/MobileProductCategoryInput" name="Incident.Product"/>
 <rn:widget path="input/MobileProductCategoryInput" name="Incident.Category"/>
 <rn:widget path="input/FormSubmit" label_button="#rn:msg:SUBMIT_YOUR_QUESTION_CMD#" on_success_url="/app/
ask_confirm" error_location="rn_ErrorLocation"/>
 <rn:condition content_viewed="2" searches_done="1">
 <rn:condition_else/>
 <rn:widget path="okcs/OkcsSmartAssistant" accesskeys_enabled="false" view_type="inline"/>
 </rn:condition>
 </form>
</div>
```

Customer Portal Customization Guidelines

Web developers who need to produce custom code to invoke Knowledge Advanced REST APIs from Customer Portal can use the OKCS API model. To use the model, you should have an understanding of web application development, PHP, software design and development methodologies, object-oriented programming, and also a basic understanding of WebDAV, HTML, CSS, and JavaScript.

OKCS API Model

The OKCS API model (**okcs.php**) provides functionality to perform Knowledge Advanced operations. It has various wrapper functions for the Knowledge Advanced REST API's and business logic to model the data for Customer Portal pages and widgets interaction. All Knowledge Advanced widgets and pages use the Knowledge Advanced REST APIs to fetch or post data.

Invoke the Knowledge Advanced REST APIs from Customer Portal

You can invoke both the Knowledge Advanced content (IM) and search REST APIs directly from Customer Portal using the OKCS API model. The OKCS API model is located in **/cp/core/framework/Models**.

The OKCS API model includes a public `makeApiRequest` method. You can write custom code to instantiate the model class, then call the `makeApiRequest` method to invoke a content (IM) or a search API. The response is stored in the `$apiResponse` variable, which can be consumed as required. You can include the custom code within a custom PHP page or a custom widget.

The `makeApiRequest` method has three parameters.

- `$url` – The API end-point URL of a REST API. Use `\RightNow\Utils\Config::getConfig(OKCS_IM_API_URL) . '<resource_URL>'` to invoke a content (IM) API and `\RightNow\Utils\Config::getConfig(OKCS_SRCH_API_URL) . '<resource_URL>'` to invoke a search API.
- `$methodName` – Method name is the type of method request, including GET or POST.
- `$postData` – Post data is an array of variable names and values in the body of the request message sent by the POST method.

You can find the complete list of Knowledge Advanced content (IM) and search resource URLs in the *Tasks* section of the *REST API for Knowledge Advanced in B2C Service* guide.

Content (IM) API Code Example

This sample code snippet shows a GET request that retrieves a document by its document Id. The request returns the knowledge base article with the specified document Id.

Example of Request Body:

```
$okcsModel = new \RightNow\Models\Okcs();
$url = \RightNow\Utils\Config::getConfig(OKCS_IM_API_URL) . 'latest/content/docId/{docId}';
$methodName = 'GET';
$apiResponse = $okcsModel->makeApiRequest($url, $methodName);
var_dump($apiResponse);
```

Search API Code Example

This sample code snippet shows a POST request that creates a question in the knowledge base repository. The request returns a set of search results for the specified question.

Example of Request Body:

```
$okcsModel = new \RightNow\Models\Okcs();
$url = \RightNow\Utils\Config::getConfig(OKCS_SRCH_API_URL) . 'latest/search/question?
question=search_keyword';
$methodName = 'POST';
$postData = '{resultLocales": "en_us", "session": null, "transactionId": 0}';
$apiResponse = $okcsModel->makeApiRequest($url, $methodName, $postData);
var_dump($apiResponse);
```

8 Configure Agent Desktop

Overview of Configuration

You must enable Knowledge Advanced in Agent Desktop so that agents can use knowledge to resolve incidents. You can enable knowledge in an existing incident workspace, in a copy of an existing workspace, or in a new workspace. You can also configure knowledge in a new chat workspace.

You enable knowledge in incident and chat workspaces by adding the Knowledge button to the workspace toolbar. You will also need to associate agents' profiles and a navigation set to each configured workspace.

You can find more information on creating workspaces, agent profiles, and navigation sets in *Using B2C Service*.

After you enable knowledge in Agent Desktop, you can configure how your agents can use knowledge in an incident workspace, including restricting access to content by user groups, displaying incident references that an article links to, enabling knowledge search by product and category, limiting locales that agents can access to, opening embedded links within Agent Desktop, and setting the article information.

Related Topics

- Video: Modifying Standard Workspaces for Knowledge Advanced

Add the Knowledge Button to the Incident Workspace

As an administrator, you can add the Knowledge button to a workspace toolbar to enable agents to access the knowledge in Agent Desktop. Agents can then click the Knowledge button on the workspace toolbar to use knowledge. You use the same process for both incident and chat workspaces.

1. Open a workspace.
2. In the main menu **Home** tab, click the **Ribbon** button to see the **Ribbon Designer** window.
3. In the **Ribbon Preview** section, click any button in the button group where you want to add the new Knowledge button.

   The **Editing Group** title bar shows the path of the button group, and the pane lists the buttons in that group.
4. In the Editing pane toolbar, click **Add Buttons**.
5. In the **Add or Remove Buttons** dialog box, check the **Knowledge** check box and click **OK** to close the **Ribbon Designer** window.
6. Log in using the profile for the workspace to verify that the Knowledge button was added.

Related Topics

- Video: Modifying Standard Workspaces for Knowledge Advanced

Add the Propose Article Button

You can add the Knowledge Advanced Propose button to the incident workspace to enable agents to propose that an incident should be the basis of a new knowledge base article. The propose option is available to only agents when they use the Browser User Interface. The propose option is available only to agents using the Agent Browser User Interface.

Agents must have the Propose permission in their user profile to see the propose option in the incident window. You can find more information on setting the propose permission in the *Service Permissions* section in *Using B2C Service*.

When agents propose an article, the application creates an article from the incident, but it remains unpublished until it is approved by a knowledge manager. Any workflow process that your organization has implemented for articles of the content type that the agent created will apply to the article.

Agents need to select the content type for the article that they are proposing, and they must select a content type that has appropriate fields so that the incident content can be properly saved as an article. If an agent selects a content type that does not have the appropriate fields, the application will not create the new article. Oracle recommends that you test this feature with existing content types and advise agents on best practice for proposing articles.

The application creates the new article by mapping the incident's fields to the article's fields as follows:

- The incident's subject maps to the text field that is the master identifier for the content type. This is usually the title.
- The first customer entry maps to the first rich text area.
- Subsequent message threads map to the second rich text area.
- The incident's products and categories are assigned to the article.
- The agent's user interface locale maps to the locale of the article.

1. Open the incident workspace.
2. In the main menu **Home** tab, click the **Ribbon** button to see the **Ribbon Designer** window.
3. In the **Ribbon Preview** section, click any button in the button group where you want to add the **KA Propose** button.
4. The **Editing Group** title bar shows the path of the button group, and the pane lists the buttons in that group.
5. Click **Add Buttons** to see the **Add or remove buttons** window.
6. Select the **KA Propose** check box and click **OK** to close the **Ribbon Designer** window.
7. Log in to the Agent Browser UI using the profile for the workspace to verify that the **Propose** option is available.

Configure the Cross-Lingual Search

The cross-lingual search feature lets an agent working in one locale supplement the search results by opening another locale. This feature relies on translated concepts in the ontology (the delivered concepts).

Concepts have synonyms in multiple languages so the application finds a synonym for a concept in other locales. The ontology is translated by default, but the customers need to translate any custom concepts they create or modify, or they won't appear in search results.

This feature is implemented and configured out-of-the-box. However, if agents can't find a specific locale when using this search, it may be restricted and you need to unrestrict it. See *Configure Users' Knowledge Locales* to restrict and unrestrict locales.

Filter Incident Search Results by Agent Role

You can limit knowledge search results in the incident workspace so that agents will see only articles assigned to the user groups they belong to. Articles that have no users groups assigned are available to all agents.

Before you can limit agents' access to articles by user groups, you need to do these tasks.

- Set up user groups.
- Create profiles and console users.
- Assign user groups to profiles.
- Assign user groups to articles.

You can find information on setting up access levels, service level agreements, profiles, and console users in *Using B2C Service*.

1. Login to the administration interface as an administration user.
2. Go to the Knowledge Advanced Configuration page in the administration interface.
3. Select **Tools** > **System** > **Configure** > **Content Visibility Configuration** to open **User Group Configuration** screen.
4. Click **Override default configuration** to enable the **Use User Group For Console User** check box.
5. Select the **Use User Group For Console User** check box, then save the configuration setting.

When agents search for knowledge in the incident workspace, they will see only articles assigned to the user groups they belong to.

Enable Searching by Product and Category

You can enable searching by product, category, or both product and category.

1. In the incident workspace, select at least one value in the **Category** and **Product** drop-down lists.
2. Go to **Site Configuration**, then **Configuration Settings**.
3. Expand **RightNow > Oracle Knowledge Cloud Services > General** folders to display the knowledge configuration settings.
4. Set OKCS_PRODUCT_CATEGORY_SEARCH to one of the following values:
    - Choose product to filter by product, if one was selected in the incident.
    - Choose category to filter by category, if one was selected in the incident.
    - Choose product_category to filter by both product and category, if either was selected in the incident.
    - Choose none to disable product and category filtering.

Display Incident References in the Article Views Page

An incident reference is a table that lists the incidents that an article links to. It is displayed on the Article View page so that agents can check whether an article has been used to solve similar incidents. The reference includes the incident number, status, summary, date and time opened, and date and time closed for each incident. Incident references display by default.

1. Go to **Site Configuration**, then **Configuration settings.**
2. Expand the **RightNow Common** > **Oracle Knowledge Cloud Services > General** folders to display the knowledge configuration settings.
3. Enable incident reference by changing the value of OKCS_SHOW_INCIDENTLINK_INFO to `true`.

Configure Users' Knowledge Locales

You can limit the locales that users can access when they use knowledge using the OKCS_USE_USER_PROFILE_LOCALE setting. The default value of the configuration setting is **No**, which allows agents to view knowledge in all of the locales defined in the repository.

If you set the value of this setting to **Yes**, users can use knowledge only in the locales specified in their user profiles, and only those locales will display in the **Language Preferences** section.

1. Go to **Site Configuration**, then **Configuration Settings**.
2. Expand the **RightNow Common** > **Oracle Knowledge Cloud Services > General** folders to display the knowledge configuration settings.
3. Set the value of OKCS_USE_USER_PROFILE_LOCALE to **Yes**.

Configure Link Documents to Display in Agent Desktop

When agents click links in the Answer Details page, the application opens the documents in Customer Portal by default. You can configure links to open in Agent Desktop instead using the OKCS_OPEN_LINK_IN_AD setting.

1. Go to **Site Configuration**, then **Configuration Settings**.
2. Expand the **RightNow Common** > **Oracle Knowledge Cloud Services > General** folders to display the knowledge configuration settings.
3. Set the value of OKCS_OPEN_LINK_IN_AD to `Yes`.

Configure Article Information on the Answer Details Page

You can set the article information that will display on the Answer Details page. This page displays article information such as document ID, version, status, published date, user group, and author. The default configuration displays the document ID, version, whether it is published, the starting date that it was or will be displayed, the name of the user who created the article, and its aggregate rating: `document_id | version | published | display_date | creator | aggregate_rating`.

1. Go to **Site Configuration**, then **Configuration Settings**.
2. Expand **RightNow > Oracle Knowledge Cloud Services > General** folders to display the knowledge configuration settings.
3. Set the value of OKCS_ANSWER_DETAILS_META_INFO to include attributes you want to display, separated by the pipe (|) character.

Attribute	Description
document_id	The article's document ID.
version	The article's version.
published	The article's status. The attribute name appears as 'Status' on the Answer Details page.
display_date	Displays the published date of article. The attribute name appears as 'Modified Date' for draft articles and 'Published Date' for published articles on the Answer Details page.
creator	The user who created the article. The attribute name appears as 'Author' on the Answer details page.
aggregate_rating	The aggregate rating of article.
owner	The article's owner.
answer_id	The article's answer ID.
last_modified	The date that the article was last modified.
last_modifier	The user who last updated the article.

Attribute	Description
user_group	The user groups that have access to the article. A user must belong to one of the user groups in order to access the article.

Enable Suggested Searches

You can enable suggested searches so that users can see a list of articles that match the search term or phrase as they type in the search field. You enable suggested searches to provide a faster way for your users to locate and select relevant articles.

Suggested searches is enabled by default in the current implementation. If you are using a previous implementation, you can enable suggested searches using the `OKCS_SUGGESTED_SEARCHES` configuration setting.

Before you enable suggested searches, you must enable attribute level searching for the master identifier (title field) of each content type in the knowledge base. You can find more information about enabling attribute level searching in the *About Schema Attribute Options* section of *Administering Knowledge Advanced* .

1. Go to **Site Configuration**, then **Configuration Settings** to see **Configuration Settings Editor**.
2. Expand the **RightNow Common** > **Oracle Knowledge Cloud Services > General** folders to display the knowledge configuration settings.
3. Set the value of **OKCS_SUGGESTED_SEARCH** to **Yes**.

Disable Contextual Search in the Agent Browser User Interface

You can configure the Agent Browser UI so that automatic contextual search is disabled, and agents have access only to manual searching when working with incidents. When you disable contextual search, knowledge ignores the contextual information from the incident, such as the subject, product, and category information. Agents must instead enter their search criteria manually using the Search the Knowledge Base tab.

You disable contextual search by turning off recommended answers in the Agent Browser UI. The current implementation contains the recommended answers tab by default. You disable recommended answers using the `OKCS_RECOMMENDED_ANSWERS` setting. The default value is 1.

1. Go to **Site Configuration**, then **Configuration Settings**.
2. Expand the **RightNow Common** > **Oracle Knowledge Cloud Services** > **General** folders to display the knowledge configuration settings.
3. Set the value of **OKCS_RECOMMENDED_ANSWERS** to 0.
4. Log in using the profile for the workspace to verify that the recommended answers tab does not display in the knowledge panel.