Oracle Digital Experience for Communications

Implementing Launch Experience

April 2024

Oracle Digital Experience for Communications
Implementing Launch Experience

April 2024

F77899-08

Author: Iqtidar Hussain

Contents

Get Help

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

Get Help in the Applications

Use help icons (?) to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons.

Get Support

You can get support at *My Oracle Support*. For accessible support, visit *Oracle Accessibility Learning and Support*.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

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

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program*. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

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

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

Thanks for helping us improve our user assistance!

1 About This Guide

Audience and Scope

This guide is for product managers, marketing managers, technical specialists (pricing specialists), and product catalog administrators, who intend to use Launch Experience to centrally define product offers, manage product lifecycles, product approvals, and raise business events which can be subscribed by a third-party application.

The guide also provides information to technical specialists and product administrators to set up Launch Experience and its related systems, introduces and explains the various components supported by Launch Experience, and further extends the concepts, such as creating product specifications or importing external catalogs, based on business needs.

How to Use This Guide

The chapters represent the sequence in which you must implement the Launch application. You must not move to the next chapter before completing the activities of the previous chapter. You may notice that some of the sections in this guide are for your reference

To implement the Launch application, you may need to refer to documentation links included in several sections of this guide. These references enable you to go to the latest information in that specific area.

Related Guides

Refer to the following guides on Oracle Help Center for additional information related to your implementation tasks.

Reference	Description
Oracle Digital Experience for Communications Using Launch Experience	Describes how you can create, publish, and manage product offers.
Oracle SCM Cloud Using Product Master Data Management	Describes user tasks for managing product catalogs, specifications, relationships, structures, and versions. Also describes importing and exporting product data and ensuring its quality.
Oracle Cloud Extending Oracle Cloud Applications with Visual Builder Studio	Describes how to use a web-based visual development tool to extend Oracle Cloud Applications.
Oracle Applications Cloud Implementing Applications	Describes setup and maintenance tasks that are generic and apply to some or all product families.

Reference	Description
Oracle Applications Cloud Using Common Features	Describes how to do general tasks that apply to some or all product families.
Oracle Communications Billing and Revenue Management Installation Guide	Describes the system requirements and procedures for installing and configuring the Oracle Communications Billing and Revenue Management system.
REST APIs for Oracle Digital Experience for Communications Launch Experience	Provides the REST API reference document for Launch Experience. The document is available on My Oracle support.
Implement CX Industries Framework	Describes the setup and implementation of the CX Industries Framework required to deploy Launch Experience.
Integrate Launch with Siebel CRM	Describes the setup and integration of Launch Experience with Seibel CRM.
Integrate Data Driven Experience in Launch	Describes the integration of third party analytics tool with Launch. It uses integration of Flytxt (a third party analytics tool) as an example.

2 Implementation Overview

Get Started

Use this topic to verify provisioning and get started with setting up the Launch application.

Verify the following:

- You're provisioned with the environment and have received an e-mail from Oracle with your environment and initial sign-on information.
- If you're assigned as the administrator, check the details of your password and instructions on how to access the environment.

When you sign in for the first time, reset your password by clicking your user image or name in the global header and selecting the Set Preferences from the Settings and Actions menu.

If you need more information on the provisioning workflow of Cloud applications, see *Workflow to Provision Oracle Cloud Applications*.

Tasks Summary

Use this topic for a summary of the setup tasks required to run the Launch application and where you would do each of these tasks.

Task	Description	Application Area
Set up users and roles	Create users and assign the roles with privileges required for Launch Experience.	Navigator > Tools > Security Console
Set up Launch	Set up tasks to use the application capabilities, such as search, content and integration to spoke systems.	• Navigator > My Enterprise > Setup and Maintenance • Navigator > Tools > Scheduled Processes
Create catalog entities	Create catalog entities through APIs or the Launch UI.	• REST APIs for Oracle Digital Experience for Communications Launch Experience on My Oracle Support. • Using Launch Experience guide
Set up catalog import or export	Import and export catalog entities.	• Launch Application UI • REST APIs
Publish catalog entities	Use initiatives to publish catalog entities to spoke systems.	• Implement CX Industries Framework, on My Oracle Support

Task	Description	Application Area
		- REST APIs for Oracle Digital Experience for Communications Launch Experience on My Oracle Support
Configure and extend	Configure Launch for use.	- Application Composer - Navigator > My Enterprise > Setup and Maintenance - VB Studio

3 Set Up Users and Roles

Set Up Initial User

Use this topic to set up the initial user for the Launch application.

After you have signed up with your Oracle cloud service, you receive the user name and password for one initial user. The initial user is provisioned with the job roles and privileges necessary to perform many implementation tasks, including creating other users.

> **Note:** You either do additional set up for the user in the Welcome Email or create another user and then grant administrative privileges to the new user.

You must add the following job roles to the initial user through Security Console. If you need more information about adding roles, see *Assign Roles to an Existing User*.

Here are the quick steps to adding roles as per your business requirements:

1. In the Security Console, click the **Users** tab.
2. Search and select the user to whom you want to assign these job roles.

Job Roles	Description
Product Manager	Manage items and item catalog, and view communications item schema.
Application Implementation Consultant	Manage users and roles, functional setups user, resource organization administration, inventory management, and elastic search setup used in the landing pages and its initial ingest.
Product Data Steward	Upload data for item import, product hub administration, and item management.
IT Security Manager	Perform user and role setup.
Communications Catalog Administrator	Manage operations for catalog resources including import, export, and publishing of catalog definitions to run-time applications.
Communications Technical Specialist	Manage technical portfolio, including creating and managing specifications.
Marketing Manager	Manage launch of new product offerings by cloning or reusing existing products, services, and more.

3. On the User Account Details page, click **Edit** .
4. In the Roles section, click **Add Role**.
5. Search for the role that you want to assign to the user and then click **Add Role Membership**.

6. Repeat the previous step to add more roles, and then click **Done**.

If you're looking for information on provisioning and activating Oracle Cloud services, adding other users who would manage the services, or about securing your Cloud applications, see *Guidelines for Configuring Security in Oracle Applications Cloud*.

Create Users and Assign Job Roles

Use this topic to set up users in the Launch application based on your requirements.

Create Application Users

Here's the mapping of users and roles that you may create using Security Console. For more information on role mappings and provisioning, see *Role Mappings*.

Create these users	Assign these roles
Administrator	Communications Catalog Administrator
Product Manager	Communications Catalog Product Manager
Marketing Manager	Communications Marketing Manager
Read-Only User	Communications Catalog Viewer
Technical Specialist	Communications Technical Specialist

Create a Resource User

You must create a resource user for the offer approvals to work. You then configure the resource hierarchy so that the approvals traverse up through the hierarchy. See *Example of a Sales Resource Hierarchy* to configure the resource hierarchy.

Do remember that the approval process would work only when the associated initiative is configured for approvals.

You must add the required job roles to the resource user through the Security Console. If you need more information about adding roles, see *Assign Roles to an Existing User*.

Synchronize Roles and Privileges

Use this topic to synchronize the roles and privileges with Security Console.

After configuring roles and users, run the **Import User and Role Application Security Data** scheduled process to synchronize the changes with Security Console. If you have administrator privileges, here are the quick steps to help you get started. For more information on scheduled processes, see *Import Roles and Privileges into Security Console*.

1. Go to **Navigator > Tools > Scheduled Processes** and then click **Schedule New Process**.
2. Click **Search** corresponding to the Name drop-down list.
3. Under Search and Select, enter **Import Users and Role** in the Name field and click **Search**.
4. From the search result, select **Import User and Role Application Security Data** and click **OK**.
5. Click **OK**, and then click **Submit**. It may take a few moments for the process to complete.

Synchronize All Users Between Launch Experience and Customer Experience Industry Framework Identity Management

To synchronize all users between Launch Experience and Industry Framework Identity Management, you need to:

- Configure the application and define user name suffixes.
- Test the configuration.
- Activate the synchronization process.

To configure the application

1. Log in to your CX Industry Framework identity domain in your Oracle Cloud Infrastructure account. You can get this link from your welcome email. If you have questions about which tenancy and domain to log in to, contact your Oracle Support team.
2. Create a new application by selecting **Applications** in the navigation pane, and then click the **Add application** button.
3. Select Application Catalog and then click the Launch app catalog button.
4. Search for and select the template named Oracle Fusion Applications Release <X>, where the release is 13 or later.
5. On the Add Oracle Fusion Applications Release screen, complete these fields:
   - **Name**
   - **Description** (optional)
   - **Application icon** (optional)
6. Click **Next**.
7. In the **General** section, complete these fields using a bogus URL that begins with **http://** and ends with **.com**:
   - **Entity ID:** http://bogus-url.com
   - **Assertion Consumer URL:** http://bogus-url.com
8. Under **Additional configurations**, complete these fields with the same URL:
   - **Single Logout URL:** http://bogus-url.com
   - **Logout Response URL:** http://bogus-url.com
9. Click **Next**.
10. Turn On **Enable Provisioning** and click **Confirm**.
11. In the **Configure connectivity** section, complete the following fields:
    - **Administrator Username:** Enter your Fusion applications administrator credentials.
    - **Administrator Password:** Enter your Fusion applications administrator password.

- **Host Name:** Enter the Fusion application URL hostname portion without **http://**. For example, **myFAhostname.oraclecloud.com**
- **Port Number:** 443
- **SSL Enabled:** Select this option.

12. In the **Provisioning Operations** section, complete these fields:
    - **Authoritative sync:** Select this option.
    - **Create account:** Select this option.
    - **Update account:** Select this option.
    - **Deactivate account:** Select this option.
    - **Delete account:** Deselect this option
13. Turn on **Enable Synchronization**.
14. Scroll up to view the **Configure Attribute Mapping** section and click the **Attribute mapping** button.
15. On the Attribute mapping screen select the **Application to identity domain** option.
16. Locate the row with the **User** column value set to **Federated** and modify the source value in the left column to be **true** where it says **false**.
17. Click the **Save changes** button, which returns you to the previous screen.
18. In the **Configure synchronization** section, complete the **Synchronization Schedule** field with the frequency you want to use for synchronization. The recommended value is **Every hour**.
19. Click **Finish**.
20. When you are ready to either test the synchronization, or make the synchronization live, Click **Activate** and continue to the next task.

After you configure the application you need to import users and groups that you want to synchronize, and test the synchronization setup to ensure that the selected application users and groups are being synchronized to the Fusion application identity domain. When you have successfully tested the synchronization, you then activate the process using the instructions in the next task.

To test the synchronization

1. From where you left off in the previous task, scroll down to the **Resources** section in the navigation panel and select **Import**, and then click the **Import** button.
2. The message on the screen indicates that the import job has been submitted and is running. Refresh the screen until the **Import status** changes to **Complete**.
3. Go back to the main screen for the Fusion applications identity domain to verify that users were successfully copied from Fusion applications.
4. In the navigation pane, click **Users** and **Groups** respectively to verify:
    - **Groups:** Verify that the groups you expect to see are available.
    - **Users:** Verify that the users you expect to see are available and that they are members of the correct groups.
5. Remove the test results by completing these steps:
    - Deactivate the application created in the previous task.
    - Delete all users and groups that were migrated into Fusion applications identity domain.
6. Complete the steps in the next task to activate the synchronization process.

To activate the synchronization process

1. Log in to your CX Industry Framework identity domain in your Oracle Cloud Infrastructure account. You can get this link from your welcome email. If you have questions about which tenancy and domain to log in to, contact your Oracle Support team.
2. Select **Domains**, then click on the domain name.
3. Select **Oracle Cloud Services** from the navigation panel and locate the application corresponding to the CXIF instance. The name starts with either **CXIF** or **DX4C**, and the description likely reads *CXIF IDCS Application*. It was created during the CX Industry Framework provisioning process.
4. Select the application and then, under **Resources**, select **Application roles**.
5. Verify that the application has the following application roles:
   - dx_DX4C_Configuration_Endpoint_Read
   - dx_DX4C_Configuration_Endpoint_Write, and others
6. Using the steps in the previous task, activate the application and import the users and groups again.
7. When the import is complete, return to your CX Industry Framework identity domain, select your domain, and then select **Groups** from the navigation pane. Verify that these groups are displayed:
   - Communications Customer Service Administrator
   - Communications Customer Service Manager
   - Communications Customer Service Representative
8. Return to the application referenced in step 3, and then select **Application roles**. The roles beginning with "dx" are displayed.
9. Assign Groups to the role **dx_DX4C_Configuration_Endpoint_Read**. To assign the groups, complete these steps for each role:
   a. Click on the action menu and click **Assign groups**.
   b. Select the three groups listed above that are associated with the utility customer service agent, manager, and administrator, and click **Assign**.
10. Once all of the groups are assigned, you have completed the process.

Create Aftermarket Extensibility Administrative User

Use this topic to create an aftermarket extensibility administrative user.

You can now extend the product offering entity by specifying a list of fields to be extended using the Launch user interface. The administrative user for this function can upload the spreadsheet containing the list of fields to be extended. The fields should be simple attributes of type text, number and check box.

For example, if a communications service provider wants to extend the product offering entity, say, SupremoProductOfferingInfo with the additional field **partnerBrand** and similar other fields, the user with the custom job role **Catalog Extension Management Duty Role** has the privileges for this extension.

Here's how you can create the user for this role:

1. In the security console, click **Roles**.
2. Click **Create role**.
3. Create a CRM job role by entering a unique role name and role code.

4. Go to the **Role Hierarchy** tab and search for **Catalog Extension**.
5. Select the **Catalog Extension Management** duty role.
6. Verify the function security policies listed in the tab.
7. Ensure that these are configured in the above list:
   - **Manage Extensible Object**
   - **Manage Catalog Extension**
   - **View Catalog Extension**
   - **View Extension Tile**

   > **Note:** The **Administer Sandbox** privilege must be added manually from **Add Function Security Policy** even if it's inherited from the **Catalog Extension Management** duty role.

8. Click **Add Function Security Policy**, search for the **Administer Sandbox** privilege and click **Add Privilege to Role**.
9. Go to the **Users** tab, and click **Add User**.
10. Select the user you want to configure the role for.
11. Click **Add user to Role**.
12. Save the job role.

4 Set Up Launch Experience

Before You Begin

This section describes the additional setup tasks you can do before implementing Launch.

Set Up VB Studio

You can set up VB Studio to extend Oracle Cloud applications. An application extension is an artifact that enables you to extend certain Oracle Cloud applications to meet your business needs. You deploy an application extension to an Oracle Cloud applications instance. Only one VB Studio instance can be provisioned in an Oracle Cloud account.

By default, a VB Studio instance is provisioned with your first Oracle Cloud Applications `*-TEST` instance that your organization's members can use to develop application extensions. If you need to use VB Studio with a different Oracle Cloud Applications instance, file a service request.

Oracle deletes the VB Studio instance associated with your `*-TEST` instance and creates a new VB Studio instance in your preferred identity domain. For more information, see *Set Up VB Studio*.

Define Your Enterprise Structure

Based on other Oracle Cloud applications that are being provisioned, you can define your enterprise structure, such as, Legal Entity, Business Unit, Primary Ledger and Chart of Accounts among others.

For more information, see *Enterprise Structures Business Process Model*.

The following table lists the mandatory sections you need to set up before you begin implementing Launch.

Section	isMandatory
Set Up Default Item Organization	Yes
Set Up Profile Options	Mandatory profile options are documented in the section.
Set Up Search Index	Yes
Set Up Search Tags in Offer	No
Set Up Oracle Content Management	Yes
Set Up Regional Values	No
Set Up Business Configuration Values	Yes
Enable Auditing in Sales Cloud	No

Section	isMandatory
Set Up Design Time Restrictions Through Entity Profiles	No

Set Up Default Item Organization

You must set up a default item organization for Oracle SCM Item Sync. You can integrate with Oracle SCM by implicitly creating a Product Information Management item for every product offering created in Launch Experience.

To set up a default item sync to Oracle SCM, you must perform the two following steps:

1. Set Up Item Organization in Oracle SCM
2. Configure Item Organization for Default Item Creation

Here's how you can do it.

Step 1: Set Up Item Organization in Oracle SCM

You must be a user with the Application Implementation Consultant role to be able to view and perform the list of tasks required to set up the item organization.

To set up an item organization:

1. Go to **Navigator > My Enterprise > Setup and Maintenance**.
2. In Setup and Maintenance, click **Tasks > Search** and enter **Manage Item Organizations**.
3. On the Manage Item Organizations page, Search Results section, click **Create** from the Actions drop-down list.
4. On the Create Item Organization page, select the **Create new** check box.
5. Enter information for mandatory fields such as Name and Organization and provide additional details such as Management Business Unit, Primary Ledger, Legal Entity, Legal Addresses, if already setup. If not, you may leave out the additional fields.
6. Click **Next**.
7. On the Manage Item Organization Parameters page, select your item master organization that you provided earlier from the Item Master Organization drop-down list.
8. Click **Save and Close**.

Step 2: Configure Item Organization for Default Item Creation

Now that the item or inventory organization creation is complete, the same must be configured in the root item class security to allow item creation against this item or inventory organization.

To set up item class security for the new item or inventory organization:

1. Go to **Navigator > My Enterprise > Setup and Maintenance**.
2. In Setup and Maintenance, click **Tasks > Search** and enter **Manage Item Classes**.
3. On the Manage Item Classes page, select the Root Item Class and click **Edit**.
4. On the Edit Item Class page, go to the Templates and Formats subtab.
5. On the Item Templates subtab, select any template, for example, Finished Goods template, edit and make it the default.
6. For the selected template, update the following attributes:
    a. Item Status: Active
    b. Lifecycle Phase: Production

   c. Primary Unit of Measure
   d. Positive Deviation Factor
   e. Negative Deviation Factor

7. Click **Save and Close**.

Set Up Profile Options

A profile is a set of configurable options that affect the way your application looks and behaves. As a system administrator, you can control how your application operates by setting the profile options to the required values.

Here's how you can go about it:

1. Go to **Navigator > My Enterprise > Setup and Maintenance**.
2. In Setup and Maintenance, click **Tasks > Search**, and enter **Manage Administrator Profile Value** in the Search field.
3. On the Manage Administrator Profile Value page, search for the profile codes as provided in the table and set the Site Profile Value for each of the profile codes.
4.

Profile Option Code	Description	Is Mandatory?
ORA_ATC_DEFAULT_ORG_CODE	Organization code for creating an item in SCM for the Product Offers created in Launch.	Yes
ORA_ATC_EXPORT_IN_MEMORY_COUNT	Used for the export functionality to define the number of objects that can be kept in memory. The default value is 1000 that you may change if required.	No
ORA_ATC_IMPORT_THREAD_COUNT	This profile option determines the number of parallel threads that run during the import. The default value is 10. Criteria to configure the profile option: ○ Number of ESS servers configured on the instance: You can get this information by contacting Oracle Support. ○ Other active processes: Other application processes that may be simultaneously running on this server and utilizing the same ESS servers.	No

Profile Option Code	Description	Is Mandatory?
	Note: If the Launch Experience import is the only process running on the instance, you can configure the number of threads up to 10 times the number of ESS servers. If there are other processes likely to be running simultaneously, reduce the number of threads proportionally.	
ORA_FND_SEARCH_EXT_ENABLED	Validate if this has been set.	Yes
ORA_ATC_LAUNCH_APPROVAL_ VOTE_ PERCENTAGE	The percentage range is between 0 and 100. Defines the voting percentage required to mark the request as approved during parallel approvals. The default is 100 percent, which means all approvals are required.	No
ORA_ATC_LAUNCH_APPROVAL_TYPE	Defines the approval type for offers. The default is parallel. ○ Parallel ○ Sequential **Note:** If you define the approval type as sequential, you must not override the default approver during offer confirmation. Doing so will change the approval type to parallel.	No

For more information on creating and setting profile option values in the Setup and Maintenance work area, see *Create and Edit Profile Options*.

Set Up Search Index

Use this topic to set up elastic search on the landing page of top-level entities.

You must run the Oracle Enterprise Scheduler job to create the index in the Search instance configured in the environment. The Oracle Enterprise Scheduler job creates the index in Search instance and also performs the initial ingestion. The records which are already created will be ingested once the Oracle Enterprise Scheduler job is completed.

Here are the details of the Oracle Enterprise Scheduler job:

- Oracle Enterprise Scheduler job name: ESS job to create index definition and perform initial ingest to OSCS.

- Index name parameter: `fa-crm-productcatalog`.

Ensure that the Oracle Enterprise Scheduler job is completed successfully. If the job fails or the elastic search doesn't work, check for the search instance availability. To check the search instance availability, run the Oracle Enterprise Scheduler job named ESS process to check Search Cloud Service availability.

You can set up elastic search for the following entities:

- Initiative
- Product Specification
- Product Offering
- Catalog
- Price List
- Promotion
- Rules
- Product Line

For information on how to run scheduled processes, see *What are scheduled processes?*.

Set Up Search Tags in Offer

Use this topic to set up search tags for an offer with keywords used to discover offers by type, specialty or domain.

Setting up search tags improves search capability over the existing one. Keywords provide relevance to the search and in turn improve the user experience, especially for non-technical persons.

Here's how to set up a search tag:

1. Click **Navigator** > **My Enterprise** > **Setup and Maintenance**.
2. Search for the **Manage Standard Lookups** task.
3. Search for the lookup code **ORA_ATC_SEARCH_TAGS**.
4. Add the values **5G**, **BYOD**, **Christmas** and so on.

> **Note:** Use the same **Lookup Code** and **Meaning** values for **ORA_ATC_SEARCH_TAGS**.

Set Up Oracle Content Management

Use this topic to set up Oracle Content Management (OCM) as the content repository for the Launch application.

Here's how you set up OCM:

1. File a Service Request for Cloud Ops to perform the following setup:
    a. Enable SSO between Launch Experience and OCM:

    > **Note:** You can skip this step if both instances are on the same IDCS stripe.

        i. SAML Integration between Launch Experience UI and OCM UI.

        ii. OAuth integration to enable Launch Experience to OCM REST API calls.
    b. Enter the OCM Connection URL in Topology Manager either via EM UI or WLST command `update_topology.py`:
        i. Enterprise App Short Name (Code): ORA_HYBRID_OCE_APP.
        ii. Complete URL.
2. Set up asset repository in OCM to manage digital assets and content items in a group.
    a. Create an asset repository to be used for Launch Experience digital content uploads. For instructions to create an asset repository, see the section on managing repositories in Oracle Cloud Managing Assets with Oracle Content and Experience guide.

       It is recommended to publish your uploaded digital content before using it for offer design.
3. Reset the default Referrer-Policy.
    a. Open the `httpd.conf` file using either a text editor or the Advanced Server Configuration page in Fusion Middleware Control.
    b. Comment the line *Header always set Referrer-Policy "same-origin"*
    c. Restart the Oracle HTTP Server after this change.

   For more information, see *Getting Started with Oracle HTTP Server*.

Set Up Regional Values

Use this topic to specify the ledger values for product offer price. As an administrator, you must complete this setup in the application to enable pricing for a region.

Set General Ledger ID Values

To enable pricing, obtain the required regional values for Money, Tax, and General Ledger attributes.

1. Sign in as an Administrator.
2. Go to **Navigator > My Enterprise > Setup and Maintenance**.
3. In Setup and Maintenance, click **Tasks > Search**, and enter **Manage Standard Lookups** in the Search field.
4. Search for the lookup type provided in the following table and add the regional values.

Lookup Type	Description
ORA_ATC_GLID	General ledger identifier for the region. If a ledger is already set up through the task Manage Primary Ledgers, update the same ledger ID.

Set Up Business Configuration Values

Use this topic to set up business configuration values for the product offer price.

You can configure business setups that are common across design time and runtime systems to ensure the eco system setups are defined once and published into target systems. This facilitates frictionless and agile customer onboarding, and allows you to centrally introduce and manage your common business setups rather than doing it in each and every runtime system. Additionally, this capability reduces configuration efforts, aids in agility, and brings in consistency across the enterprise.

The seeded configuration items are documented below. All the configurations can be updated and extended by customers.

Configuration Characteristics	Default Configuration Characteristics Value	Description
Customer Type	Existing Subscriber New Subscriber	Customer Type values are used as parameters while defining eligibility rules.
Pricelist Type	Business Residential	Price list Type values are used to classify price list.
ELIGIBILITY_PARAMETER	Channel Code Country Customer Type Pricelist Type Market Segment Code Retail Store Code Account Type Credit Score Black List	Eligibility parameters values that are permissible to define eligibility rules.
ELIGIBILITY_OPERATOR	Contains Equals Not contains Not Equals Not starts with Start with In Is null Between Less than Less than or equal to Greater than Greater than or equal to	Eligibility operator values that are permissible to define eligibility rules.

Configuration Characteristics	Default Configuration Characteristics Value	Description
Country	No default value. Can be updated with any standard country code defined in Fusion.	Country code values used to define eligibility rules. Country code values used to define eligibility rules. See the list of available countries with loqate geography reference data in the Oracle Help Center link below the table. Geography validations are triggered while creating the common business configurations and using it in the promotion. • The country code and various geographical levels of the country are validated. • The promotions **criteriaPara** and **criteriaValue** are validated with the common business configurations and geographical levels.
Account Type	No default value. To be setup by the customer.	Account type values are used as parameters to define eligibility rules.
Credit Score	No default value. To be setup by the customer.	Credit score values are used as parameters to define eligibility rules.
Black List	No default value. To be setup by the customer.	Black list values are used as parameters to define eligibility rules.

As mentioned above, geography validations are triggered while creating the common business configurations and using it in the promotion. However, you need to configure the country structure and hierarchy in fusion applications using the **Manage Geographies** task in customer data management. See *How do I manage Geography Structures, Hierarchies, and Validation?*.

For a list of available countries with loqate geography reference data, see *List of Available Countries with Loqate Geography Reference Data*.

> **Note:** Existing default configuration values can only be disabled or enabled. However, new values can be added, enabled and disabled.

Enable and Configure Auditing for Launch Entities

Use this topic to configure auditing for Launch entities.

Auditing helps answering three specific questions - Who, what and when. Who took an action, on what data and when was it done. Auditing is about providing a history or chain of actions for any business data you work with - create, update, delete, export, import, or publish.

You need to enable auditing in Sales Cloud before configuring it for Launch entities.

Here's how you can enable auditing in Sales Cloud:

1. Click **Navigator** > **My Enterprise** > **Setup and Maintenance**.

2. Search for the **Manage Auditing Policies** task.
3. Set **Audit Level** to **Auditing** for Oracle Fusion Applications.
4. Save and close.

After you are done, you can record audit information on product specifications or product offerings. An audit log stores detailed change history information and can also be downloaded. It contains information about an action taken and the changes made to the specification - the date and time of the action, and the identity of the person taking the action.

Here's how you can configure auditing for an entity in Launch:

1. Go to **Administration** > **Audit trail** > **Manage auditing policies**.
2. In **Audit Configurations**, select either Product Offering or Product Specification as the entity type.
3. In the **Audit Status** page, select the field resources or the sub resources you want to audit in the entity type.
4. Click **Update**.

Set Up Design-time Restrictions Through Entity Profiles

Use this topic to set up design-time restrictions (supporting run-time capabilities) through entity profiles.

The modelling capabilities of design-time and run-time applications are common regardless of whether you publish to Oracle or non-Oracle run-time applications. However, there could be a gap between what could be modeled at design time as against what the application would actually use at run-time, depending on the service provider's ecosystem. You could then set modeling rules for design-time features available for a service provider, ensuring the design-time application follows the applied restrictions resulting in an error-free publishing to runtime applications. Such restrictions are based on the service provider's ecosystem and not on individual runtime applications. Which means, there could only be one restriction for the ecosystem and not for each runtime application. You can achieve this by setting up an entity profile.

Using the following two examples, you can set up an entity profile and follow some best practices:

1. Example 1 - **Rule**: Restrict only one commitment term for the product offering entity of package type.
2. Example 2 - **Rule**: Restrict one-time price.

Here's how to do it using **Example 1**.

Step 1: Create an entity profile of type customProfileSpecification

To create an entity profile:

1. Define a valid ID and name to create the entity profile.
2. Set the version to **1.0**.
3. Set the **LifecycleStatus** to **In design**.
4. Set the **@type** to **CustomProfileSpecificationOracle**.
5. Set the **profileType** to **ENTITY_PROFILE**.

Target product schema should be based on the entity restriction, such as, for productOfferingOracle, the **@type** value should be **ProductOfferingOracle** and the **@schemaLocation** value should be **ProductOfferingOracle.yml**. Similarly, you need to update type and schemaLocation values for other entities as well (**ProductSpecificationOracle**, **PromotionOracle** and so on).

Step 2 : Create a restriction based on a condition

To create a restriction:

1. Define a rule with **characteristicType** as **RULE** and **relationshipType** as **AGGREGATION**.

2. Based on the name given in **AGGREGATION**, create a **FEATURE** characteristic (`rule1_info_productType_bundle`).
3. If the restriction has a condition, you need to define that condition in the feature characteristic with **customProfileSpecCharRel** as **relationshipType** with the value **CONDITION**.
4. Create **ATTRIBUTE** based on the name given in the CONDITION type (`condition_productType_bundle_package`).
5. In **ATTRIBUTE**, define the attribute name and value to satisfy the condition (attribute name defines the condition expression).
6. Once the condition is satisfied, the respective restriction attribute (`requires_term_type_commitmentTerm`) is executed.

Here's how to do it using Example 2.

Step 1: Create an entity profile of type customProfileSpecification

To create an entity profile:

1. Define a valid ID and name to construct the entity profile.
2. Set the version to **1.0**.
3. Set the **LifecycleStatus** to **In design**.
4. Set the **@type** to **CustomProfileSpecificationOracle**.
5. Set the **profileType** to **ENTITY_PROFILE**.

> **Note:** Target product schema should be based on the entity restriction. For example, for productOfferingOracle, the **@type** value should be **ProductOfferingOracle** and the **@schemaLocation** value should be **ProductOfferingOracle.yml**. Similarly, you need to update type and schemaLocation values for other entities as well (**ProductSpecificationOracle**, **PromotionOracle** and so on).

Step 2 : Create a restriction without conditions

To create a restriction:

1. Define a rule with **characteristicType** as **RULE** and **relationshipType** as **AGGREGATION**.
2. Based on the name given in **AGGREGATION**, create a feature characteristic with **relationshipType** set as **EXCLUDES**, **REQUIRES** or **ALLOWED**.
3. Based on the name given in the feature characteristic, create **ATTRIBUTE**.

To see sample payloads for creating restrictions, see REST APIs for Launch Experience on My Oracle Support, *Doc ID 2725039.1*

To view a list of design-time restriction properties and values used to create restrictions, see *Design-time Restriction Properties to Create Entity Profiles* in the Appendix.

5 Create Catalog Entities

Use REST APIs to Create Catalog Entities

Use this topic to understand the REST endpoints that you can use to create and manage catalog entities using REST APIs.

> **Note:** For information on the Launch REST APIs, see REST APIs for Launch on My Oracle Support, *Doc ID 2725039.1*.

The default API language provided to you is English. However, you can select another language before you create the catalog entities. For example, if you choose Korean, set the language header as **Accept-Language: KO**. See *Set General Preferences for All Users*.

Manage Lifecycle Configurations

You can create and configure lifecycle statuses in addition to the ones that are already shipped with the application but you can't change the behavior of the seeded statuses. For more information, see *Manage Entity Lifecycle Using Initiatives*.

Available REST endpoints are create, create or update, delete, get a lifecycle configuration, get all lifecycle configurations, and update a lifecycle configuration.

Manage Initiatives

You must use initiatives to group catalog entities and then publish the initiative to spoke systems. An initiative enables you to group and test all your product offers, activate them, and then finally launch these entities. Referencing an initiative is mandatory when creating catalog entities.

Available REST endpoints are create, create or update, delete, get one, get all, and update an initiative.

For information on how you work with the initiatives from the Launch UI, see *Manage Initiatives*.

Manage Balance Element

Balance element is the unit of measure for elements whose usage is measured. It can be of type currency or non-currency and specifies the rounding mode, consumption rules, decimal precision, and so on.

You must add balance elements before you start to create a price list. You must also create a balance element with currency type. A currency balance element is used in creating price lists and setting up prices on offers.

You can additionally create noncurrency balance elements which can be used to configure allowance resources used in setting up prices on offers. For the balance element payload and other Launch REST APIs, see REST APIs for Launch Experience on My Oracle Support, *Doc ID 2725039.1*.

Available REST endpoints are create, create or update, delete, get a balance element, get all balance elements, and update a balance element.

Manage Price Lists

A price list is a set of standard prices for products and services. You can use multiple price lists to offer separate prices for the same product.

You must have the price list set up to create offers with pricing models. You do this using the Launch user interface as follows:

- You create the price list by going to **Administration** > **Price Lists**.
- During price list creation, you specify the currency balance element created in the **Manage Balance Element** section in this topic.

You can also set up price lists using REST API end points. For information on the Launch REST APIs, see REST APIs for Launch on My Oracle Support, *Doc ID 2725039.1*.

For information on how you work with pricing in the Launch application, see *Pricing Overview*.

Manage Tax Service Provider

A tax service provider represents simple and seller specific tax codes that are used in product offer pricing. Available REST endpoints are create, create or update, delete, get a tax service provider, get all tax service providers, and update a tax service provider.

Manage a Catalog and its Categories

After you create an initiative, you must associate the catalogs entities to the initiative to publish to spoke systems. Available REST endpoints are create, create or update, delete, get a catalog, get all catalogs, and update a catalog.

After you create a catalog, you must associate the categories inside the catalogs. You provide the category details as part of the category reference in the request body. Available REST endpoints are create, create or update, delete, get a category, get all categories, and update a category. For more information on catalogs and categories, see *Create Catalogs and Categories*.

Manage Product Lines

You can now associate the catalogs and categories to product lines.

Available REST endpoints are create, create or update, delete, get a product line, get all product lines, and update a product line.

For more information on product lines, see *Product Lines Overview*.

Manage Service Specifications

Available REST endpoints are create, create or update, delete, get a service specification, get all service specifications, and update a service specification.

For information on how you work with the service specifications in the Launch application, see *Manage Service Specifications*.

Manage Usage Specifications

Available REST endpoints are create, create or update, delete, get a usage specification, get all usage specifications, and update a usage specification.

Manage Custom Profile Specifications

These are generic profile specifications with difference profile types used for various purposes.

For example, these can be used as a promotion profile specification, charging terms profile specification, suspension terms profile specification, renewal terms, common business configuration, return checklist, trade in, device specification, entity profile and so on.

You can also define the charging or suspensions terms for an offer by selecting the seeded charging or suspension term template from the Launch application UI. This enables the spoke systems to apply the same terms as configured in the chosen template during the charging or suspension scenarios. You can also create additional templates using the format of the seeded template.

While creating a device trade-in program, you would specify the Device Specification using customProfileSpecification REST API with with profileType as TRADE_IN.

You must have Communications Catalog Administrator job role to define the charging and suspension terms using the REST endpoint `customProfileSpecification`. The seeded term templates must be created as part of the implementation.

The **lifecycleStatus** must be `Active` to make these templates available in the UI. To do this, run PATCH method call on **`crmRestApi/atcProductCatalog/11.13.18.05/productCatalogReferenceManagement/v1/customProfileSpecification/{id}`** to set the lifecycle status to `Active`.

For the charging and suspension term templates and other REST APIs for Launch on My Oracle Support, see *Doc ID 2725039.1*.

Available REST endpoints are create, create or update, delete, get a customer profile specification, get all customer profile specifications, and update a customer profile specification.

Manage Customer Profile Specifications

The REST endpoints available are create, create or update, delete, get a customer profile specification, get all customer profile specifications, and update a customer profile specification.

Manage Product Specifications

You can now create your product specifications to be able to create product offers. You can also associate specifications such as service, usage and customer profile specifications to product specifications. Additionally, when you create a product specification, you can also define SKUs through REST endpoints. You can create SKU templates and SKUs for physical goods. SKU templates allow you to generate SKU numbers for the different combinations of product attributes. SKU numbers are used to group similar product definitions, such as, simple physical goods with variant attributes.

Available REST endpoints are create, create or update, delete, get a product specification, get all product specifications, and update a product specification.

For information on how you work with the product specifications in the Launch application, see *Product Specification Overview*.

Manage Product Offers

A product offer represents entities that are orderable from the provider of the catalog and includes pricing information. A product offer can be of the type simple or bundle and can be in the form of devices and accessories or a service

Available REST endpoints are create, create or update, delete, get a product offering, get all product offerings, and update a product offering.

Ensure the following when creating the request body for the REST PUT or POST operation to create a product offering:

- Provide the name of the product offering.
- Provide the ID. This is optional but recommended. The application generates this ID if you don't provide it.
- Lifecycle status defaults to In design when a new offer is created with version 1.0.

Additionally, if you need to associate terms, rules, or catalog images to a product offering, you can do that by adding attachment references in the product offering POST/PUT request body.

For information on how you work with offerings in the Launch application, see *Manage Product Offers*.

Manage Product Offer Prices

After you create a product offer, you must add a price to it by either associating an existing price from the price list or by creating a new price. Product offering price is an amount, usually of money, that's asked for or allowed when a product offering is bought, rented, or leased. Available REST endpoints are create, create or update, delete, get a product offering price, get all product offering prices, and update a product offering price.

Manage Constraints

Constraint represents a policy or rule that you want to be applied to an entity or entity specification for pricing.

Available REST endpoints are create, create or update, delete, get a constraint, get all constraints, and update a constraint by ID.

Manage Pricing Logic Algorithm (PLA) and PLA Specifications

You can define PLAs use the characteristics defined within them while creating an product offer. The PLA specifications resource lets you view, create, update, and delete PLA specifications. Available REST endpoints are create, create or update, delete, get a PLA specification, get all PLA specifications, and update a PLA specification.

A PLA is an instance of PLA specifications and uses black-box pricing algorithm to calculate the final price of the product. Available REST endpoints are create, create or update, delete, get a PLA, get all PLAs, and update a PLA specification.

Manage Promotions

Promotions are awards, discounts, or reductions provided to customers. Available REST endpoints are create, create or update, delete, get a promotion, get all promotion, and update a promotion. For more information about promotions, see *Promotions Overview*.

Manage Device Trade-In

Device trade-in programs help your customers exchange their old devices for new devices using discounts or credits. The available REST endpoints are create, create or update, delete, get a promotion, get all promotion, and update a promotion. For more information on device trade-in, see *Device Trade In*.

Manage Rules

Rules are conditions that you can set for product offers, product lines, and product specifications. You can add or find the list of operators available for rules in ORA_ATC_OPRTOR lookup type.

Available REST endpoints are create, create or update, delete, get a rule, get all rules, update a rule, and update or create rules in bulk. For more information about rules in the Launch application, see *Rules Overview*.

Manage Entitlements

Entitlements are part of the sales / service agreement which includes information about the special pricing which can be applied against the sales products. This is refereed as entitlement-based pricing.

Available REST endpoints are create, create or update, delete, get an entitlement, get all entitlements and update an entitlement.

Manage Import and Export Jobs

You can import catalogs into Launch application. Available REST endpoints are create and get an import job. Additionally, you can also export catalog entities. For the latter, the available REST endpoints are ExportJob Get By Id , ExportJob Post, and ExportJobGet. See *Import Catalog Entities*.

Manage Digital Assets

You can manage your digital assets and content files through Oracle Content Management (OCM) cloud, which is the digital assets repository for Launch Experience. Available REST endpoints are create or update a schema file, create or upload an attachment file, download file attachment, download schema file, get folder contents, upload file attachment, upload schema file. See *Set Up Oracle Content Management*.

Publish your Catalog Definitions

To be able to publish initiatives to spoke systems, you must register destinations or spoke systems that can receive the publishing events. Available REST endpoints are create, create or update, delete, get a destination, get all destinations, and update a destination.

After the destinations are set, you can now create the initiative publish event. The REST endpoint that the application provides you is: create.

For an example of a notification listener that receives the notification `ProjectPublishEvent`, see *Doc ID 2725039.1* on My Oracle Support.

You can create publishing acknowledgments after creating a publishing job. This helps you to receive the acknowledgment from the spoke system and update the publishing status corresponding to the spoke system. The available REST endpoint is create.

6 Set Up Catalog Import and Export

Import Catalogs

Use this topic to understand how you can import third-party catalogs into the Launch application.

You can import third-party catalogs either through APIs or through the UI using JSON format. It can be a ZIP file containing all the JSON file formats. All the entities imported are assigned to a single project to manage the publish process. The import process is flexible and enables you to import one or more of the entities in a sequence by determining the dependencies between the entities.

You can import catalog definitions in the following ways:

- One at a time: Enables you to import in bulk, one entity at a time. For example, a set of product specifications, product lines, simple offers, bundle offers, price lists, and so on. However, you must ensure that the referenced entities are either in the same project or already in a lifecycle status of Active or Launched. For example, you could choose to bulk import product specifications, while ensuring that its dependent service specifications, usage specifications are in place. This also applies for the hierarchy while importing product offers to have its associated product specifications in the application.
- The entire structure in one go: For example, a Package type offer along with its bundles, both commercial and service bundles, simple offers, price lists, terms, product lines and category associations.

The sequence in which the catalog entities are imported are as follows:

- Project
- Product Lines
- Catalog
- Category
- Balance Element
- Price List
- Customer Profile Specification
- Custom Profile Specification
- Promotion
- Usage Specification
- Service Specification
- Product Specifications
- Pricing Logic Algorithm Specification
- Pricing Logic Algorithm
- Pricing Constraints
- Tax Service Provider
- Product Offer Price
- Product Offer
- Product Rule

- Entitlement

Getting Ready for Import

Before you begin an import, it's recommended that you increase the number of import threads using the ORA_ATC_IMPORT_THREAD_COUNT profile value. This profile option determines the number of parallel threads that run during the import. The default value is 10.

You can use the following criteria to configure the profile option:

- Number of ESS servers configured on your instance: You can get this information by contacting Oracle Support.
- Other active processes: Other application processes that may be simultaneously running on this server and utilizing the same ESS servers.

If the Launch Experience import is the only process running on the instance, you can configure the number of threads up to 10 times the number of ESS servers. If there are other processes likely to be running simultaneously, reduce the number of threads proportionally.

Here's how you can set the profile options for import:

1. Go to **Navigator > My Enterprise > Setup and Maintenance**.
2. In Setup and Maintenance, click **Tasks > Search**, and enter **Manage Administrator Profile Value** in the Search field.
3. On the Manage Administrator Profile Value page, search for **ORA_ATC_IMPORT_THREAD_COUNT** in the profile option code.
4. Set the profile value for Profile Level to Site and change the profile value to the appropriate number.
5. Click **Save and Close**.

How to prepare data files for import through APIs and UI

As the schema of the import file is based on TMF 620, the third-party catalog import file must be prepared in TMF JSON schema with Oracle shipped extensions. The template schema has resources which are supported by the import job. Each resource is an array of records that must adhere to the seeded schemas for Launch Experience. You must use the template schema to download the seeded schema files that are referred within each resource. See *Doc ID 2725039.1* on My Oracle Support for sample template and payload.

Here is a list of some of the key attributes present in the `importJob` resource:

- `id`: Identifier of the import job.
- @type: Indicates the type of job.
- contentType: Indicates the content type of file which was used in import job.
- status: Indicates the status of triggered import job.
- `href`: Indicates the url of triggered import job.
- url: URL of the file containing the import data.
- errorLog: Indicates the error summary of triggered job.
- errorLogUrl: Indicates the error log file url. The error file will show the error messages and corresponding IDs of the failed entities with summary information. fileName: indicates the name of the imported file.
- importSummary: Indicates the success summary of triggered job in terms of each successfully imported resources and their success count.
- path: Indicates the path of job request file.
- createdBy: Indicates the submitter of the job request.

- creationDate: Indicates the time when job was triggered.
- completionDate: Indicates the time when job was completed.

What you must know

- Multiple JSON files can be archived and imported as a single ZIP file.
- The `productOfferingPrice` subresource within the `productOffering` resource only supports references to top-level `productOfferingPrice` resource. So, you must only provide the reference when associating a `productOfferingPrice`. The complete structure of price must be included in the top-level `productOfferingPrice` resource array in the same data file, if not already present in the application, when associating a `productOfferingPrice`. Lifecycle statuses are by default set to In design for all the imported resources.
- If an existing Launch Experience top-level entity is imported, then the entity is updated. If the entity doesn't exist, then a new entity will be created as version 1.0.
- If you intend to publish the model to Buying, Catalogs and Categories should be a part of the same initiative. If there are multiple initiatives, none of the reference initiatives should have categories. Initiatives must be in the **In Design** state until all reference initiatives are imported successfully and published to Buying in the same order.
- Project and Initiative are used interchangeably.
- The lifecycle status value for all the imported records is set to In design.

How to use a project for import

Every import job uses one top-level project. If the top-level project isn't provided, the import job implicitly creates a top-level project with ID Import_<Timestamp> and name Import Job <ImportJob_ID> and associates all the imported objects to this project. This top-level project gets imported as In design lifecycle status with all other import file entities associated to it as project items. The project references within all the imported objects are optional and are defaulted to the top-level project used by the import job. If project references are provided, it must refer to the single top-level project provided in the same input file.

> **Note:** For Zip file imports, the top-level project can be provided in any one of the JSON files contained in the Zip file for import. Ensure that only one file contains the top-level project.

How to initiate an import

Through APIs

You must use the TMF product catalog management import endpoint which involves the following:

- Creating payloads using the import template. For a sample schema template and payload, see *Doc ID 2725039.1* on My Oracle Support.
- Using the curl command to initiate the import process:
  - Header:
    - Content Type: Use multipart or form data
    - Authorization: Use the standard basic authorization, such as, an encoded user name and password.
  - Form: primaryFile: Use the import data file which you have prepared.

Here's a sample of the curl command that you must use to initiate the import process:

```
curl --location --request
POST https://<hostName>/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/
importJob/\
--header 'Content-Type: multipart/form-data' \
--header 'Authorization: <ID>' \
--form '<import data file>'
```

Through UI

1. Go to **Administration > Job Management > Import Jobs**.
2. Click **Create Import Job** and select the payload prepared for import.
3. Click **Import**.

For a sample request payload on initiating the import API, see *Doc ID 2725039.1* on My Oracle Support.

How to review the import status

After you have initiated the import job, you can check on the import status and error scenarios, if any, through API. The Oracle Enterprise Scheduler job status can be checked from the **Status** field of the `GetById` response. The `errorLog` attribute in `importJob` GET by ID response indicates the error summary of the triggered job and must be used to identify the failure and fix the input data file. The importSummary indicates the successfully imported records. The structure of the `ImportSummary` also includes details like total number of objects successfully imported along with the name and count of each of the individual successfully imported resources. To initiate the import job status, use the following curl command:

```
curl --location --request
GET 'https://<hostName>/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/
importJob/<ID>' \
--header 'Content-Type: application/json' \
--header 'Authorization: <ID>'
```

Here's a sample of a successful response:

```
{
 "id": 149312,
 "@type": "ImportJobOracle",
 "contentType": "application/json",
 "status": "SUCCEEDED",
 "fileName": "testPayload.json",
 "createdBy": "booth",
 "path": "",
 "url": "v1/importFile/testPayload.json",
 "errorLog": "",
 "importSummary": {
 "id": 149312,
 "totalImportObjects": 11,
 "resources": [
 {
 "name": "project",
 "count": 1
 },
 {
 "name": "productOffering",
 "count": 10
 }
 ]
 },
 "errorLogUrl": "",
 "creationDate": "2021-04-07 14:28:00.195",
 "completionDate": "2021-04-07 14:29:26.382",
 "href": "https://hostName/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/
importJob/149312"
}
```

For a successful response, the `importSummary` lists out the success summary. To verify a successful import and get summary, use the following curl command:

```
curl --location --request
GET 'https://<hostName>/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/
importJob/<ID>' \
--header 'Authorization: <ID>'
```

Here's a sample of a failed response:

```
{

 "id": 146623,
 "@type": "ImportJobOracle",
 "contentType": "application/json",
 "status": "ERROR",
 "fileName": "samplePayload.json",
 "createdBy": "booth",
 "path": "",
 "url": "v1/importFile/samplePayload.json",
 "errorLog": "Errors importing data for resource productOfferingPrice. 3 error(s) occurred.\n\nErrors
 importing data for resource productOffering. 5 error(s) occurred.\n\nErrors importing data for resource
 constraint. 3 error(s) occurred.\n\nErrors importing data for resource productOfferingPrice. 5 error(s)
 occurred.\n\nErrors importing data for resource productOffering. 6 error(s) occurred.\n, 6 record(s) rolled
 back.",
 "importSummary": {
 "id": 146623,
 "totalImportObjects": 110,
 "resources": [
 {
 "name": "project",
 "count": 1
 },
 {
 "name": "category",
 "count": 4
 },
 {
 "name": "productLine",
 "count": 4
 },
 {
 "name": "catalog",
 "count": 1
 },
 {
 "name": "pricelist",
 "count": 2
 },
 {
 "name": "productSpecification",
 "count": 13
 },
 {
 "name": "productOfferingPrice",
 "count": 46
 },
 {
 "name": "productOffering",
 "count": 39
 }
 ]
 },
 "errorLogUrl": "v1/importFile/146623.log",
 "creationDate": "2021-04-06 10:47:01.564",
 "completionDate": "2021-04-06 10:56:44.045",
 "href": "https://hostName/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/
importJob/146623"
}
```

In the failure response, the errorLog lists out the error summary as well as the total rollback records. As some records are processed successfully as well, they're indicated in the ImportSummary. The errorLogUrl to initiate the case is: **https://hostName/crmRestApi/atcProductCatalog/11.13.18.05/v1/importFile/146623.log**.

Export Catalogs

Use this topic to know the different ways using which you can export catalog entities from the Launch application.

You can export product catalog entities either through REST APIs or through UI (applicable only for initiatives) and retrieve the exported data files from a known location. The records are exported into a single JSON file or a ZIP file containing multiple JSON files, based on the number of records, which can later be used for importing into another Launch Experience environment.

> **Note:** While exporting large number of objects from Launch application, there might be a risk of increased memory usage and file size. To avoid this, use the ORA_ATC_EXPORT_IN_MEMORY_COUNT profile option to configure the maximum limit for the number of objects that can be stored in the memory. The default value is 1000.

When the number of objects in the memory reaches this specified limit, it starts dumping into files in between and clears the memory. This results in the generation of multiple files that are zipped into a single ZIP file at the end.

Here are the ways to export catalogs:

Export by Resource Types

Here are some of the primary export resource types available in the ExportJob resource. Export job allows exporting of specific resource types using the resourceType attribute. Following are some of the TMF and non-TMF object types that can be specified when submitting export jobs:

- `productoffering`
- productOfferingPrice
- productLine
- constraint
- pricingLogicAlgorithmSpecification
- promotion
- productSpecification
- serviceSpecification
- usageSpecification
- customerProfileSpecification
- customProfileSpecification
- taxServiceProvider
- `pricelist`

- balanceElement
- catalog
- category
- productRule

Additionally, here are some of the product offering resource sub-types available to you.

- package
- commercial_bundle
- service_bundle
- service
- device

The following export options are also supported for a resource-based export. However, it's only the latest version of the resource that gets exported.

- Name: Allows exporting objects in a specific resource type based on name attribute. This attribute also supports % search to export all objects based on a name pattern.
- Lifecycle Status: Allows exporting objects across multiple resource types based on a valid lifecycle status attribute.
- Last Update: Allows exporting objects filtered based on their lastUpdate attribute for a specified date range, whenever name attribute isn't used.

  ```
  "lastUpdate": {
   "startDateTime": "yyyy-MM-dd'T'HH:mm:ss.SSSZ",
   "endDateTime": "yyyy-MM-dd'T'HH:mm:ss.SSSZ"
   }
  ```

  For a sample payload with lastUpdate, see REST APIs for Oracle Digital Experience for Communications Launch Experience on My Oracle Support, *Doc ID 2725039.1*.

Export by References

You can export resources that contain the exportReferenced attribute to indicate if the references present in exported resources must also be exported from the application. This results in multiple resource types being exported, so that they can be imported into another Launch Experience environment.

Export by Project

You can export the resources associated to an initiative using the project attribute available in the export job. Initiatives in In design status aren't supported for export.

Export by File Name Prefix

You can provide a specific file name prefix for the file containing the exported objects in the fileNamePrefix attribute. The exported file's name is then set to <fileNamePrefix>_<ExportJob_ID>. If the fileNamePrefix isn't provided, then the exported file's name is defaulted to export_<ExportJob_ID>.

What you must know

- For Export by Resource Types
  - Name
    - Must be a value or pattern (with search %) already present in the application.

        - Attribute isn't supported when exporting more than one resource types.
        - Attribute isn't supported when exporting for date range specified using lastUpdate attribute.
    - Lifecycle Status
        - Must be a value configured in the application.
        - Attribute isn't supported for exact name match (without search %).
    - ResourceType values can be only one of the supported values.
    - ResourceType doesn't support a TMF object type and its corresponding sub type together when creating an export job.
    - Project attribute isn't supported.
- For Export by Project
    - Initiatives in In design status aren't supported for export.
    - The initiative name or ID is required and must be a value already present in the application.
    - The resourceType attribute isn't supported.
    - The lifecycleStatus attribute isn't supported.
    - The lastUpdate attribute isn't supported.
- For Export by File Name Prefix: The fileNamePrefix is optional but the length is limited to 30 characters.
- A Zip file is generated if there are more than 1000 objects that need to be exported.

How to initiate an export

Through API

You must use the TMF product catalog management exportJob endpoint which involves the following:

- Creating payloads using the export template. For sample schema files, see *Doc ID 2725039.1* on My Oracle Support.
- Using the REST endpoints:
    - Method: POST
    - URL: https://hostName/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/exportJob
    - Header:
        - Content-Type: Use application or JSON.
        - Authorization: Use Standard Basic Authorization, provide encoded user name and password.

Through UI

1. Go to **Initiatives**.
2. Select and View the initiative that needs to be exported.
3. Click **Export**.

   The export job is initiated and the Job ID is notified to you.

Track the job in the Administration>Export Job UI page. The Administration>Export Job UI lists all the export jobs that have been triggered. You can monitor and track the progress of the export job on the page. Once the job succeeds, the view page would have download links to download the exported content.

How to review the export status

To verify the export job status, do a GET by ID API call on the export endpoint using the following curl command and check the response field status.

```
curl --location --request GET 'https://hostName/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/
productCatalogManagement/v4/exportJob/<exportJobId>' \
--header 'Content-Type: application/json' \
--header 'Authorization: <ID>'
```

Here's the response for the curl command:

```
{
 "id": <ID>,
 "@type": "ExportJobOracle",
 "href": "https://hostName/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/
exportJob/<ID>",
 "status": "RUNNING",
 "creationDate": "2021-01-21 07:11:38.641",
 "completionDate": "",
 "exportOptions": {
 "exportReferenced": true,
 "filenamePrefix": "exp",
 "resourceType": [
 "productOffering"
 ],
 "id": "TestProductOfferId123",
 "name": "Test Product Offer"
 },
 "createdBy": "booth",
 "path": "productOffering"
}
```

In this GetByID export API response, the field called status, indicates the current job status.

For a success response, exportSummary lists out the success summary. To verify a successful export and get the summary, use the following curl command:

```
curl --location --request GET 'https://hostName/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/
productCatalogManagement/v4/exportJob/<exportJobId>'\
--header 'Content-Type: application/json' \
--header 'Authorization: <ID>'
```

You can see the exportSummary parameter in response to easily identify the summary of a successfully exported resource. The ExportSummary shows the totalExportedObjects as well as a resource-wise split up of each resource being exported.

Troubleshoot Catalog Import Errors

Use this topic to understand how to troubleshoot some of the errors that occur during Import job. These errors may occur when the input data file used for import is incorrect.

You must use the error information provided by the errorLog and errorLogUrl attributes in the importJob GET by ID response to identify and address the issue in the input data file. The errorLog attribute provides a summary of errors that has occurred in the import job, while the errorLogUrl gives an error log file URL which shows the exact resource ID, resource name, and the error message. A new import job may be submitted with the fixed data file.

Here's the list of potential errors along with some troubleshooting tips.

Error	Error Description	Troubleshooting Tips
Wrong JSON format of the input data file	The records in the input data file are based on specific JSON formatting. Incorrect and invalid formatting results in errors.	Refer to the instructions provided in the How to prepare data files for import section to resolve this error.
Wrong attributes used in the input payload resource	Use of unknown attributes which aren't part of Launch Experience entities results in errors.	You must correct the individual records to use only the supported attributes.
Records rolled back	Import runs in multiple batches. Any error in a batch results in all records in that batch to be rolled back. Sample error message with resulting rollback records in the log file: `"jobId: 102465|id: 500TextMessage|name: 500 Text Message |resource: productOffering|errorMessage: <Actual Error for this record> |status: FAILED| The following record(s) were rolled back due to 1 error(s) in the Import Batch sub job:- jobId: 102465|id: SpotifyMusic|name: Spotify Music|resource: productOffering|status: ROLLBACK| jobId: 102465|id: ModCaseClearM15M25M35|name: Moderna Case Clear - M15/M25/M35|resource: productOffering|status: ROLLBACK|"`	As a corrective action, you have to check the log file, correct the errors and rerun the import. When the error is fixed, the rollback record gets resolved by itself.
Schema validation failures	The records are validated based on the schema for base resource types for each resource mentioned in the schema template. Error messages appear when the validation fails.	You must provide the required attributes for all the resources and subresources in the correct format for attribute values.
Reference ID validation	Any reference to an existing or new top-level catalogManagement resource that isn't provided in the same input file is validated based on the ID value passed in the reference attributes. The error message appears in the following format: **`The <reference resource name> with id <reference resource id> referenced within <parent resource> with id <parent resource id> and name <parent resource name> doesn't exist in the system.`** Any reference to a new top-level catalogManagement resource that's provided in the same input file will be provided with an empty ID value. This gets validated based on the name value passed in the reference attributes. The error message appears in the following format: **`The <reference resource name> with name <reference resource name> referenced within <parent resource> with id <parent resource id> and name <parent resource name> isn't present in the input file.`**	For an existing or new top-level catalogManagement resource that isn't provided in the same input file, you have to either fix the reference ID or include the top-level record in the same input file. For a new top-level catalogManagement resource that's provided in the same input file, you have to ensure that a unique name value is provided in the top-level and reference resources. If the referred top-level project doesn't exist in the input file as a separate top-level resource, it must be added. Otherwise the incorrect reference must be corrected to refer to the existing top-level project.

Error	Error Description	Troubleshooting Tips
	If the mandatory project reference in any of the resource is different from the top-level project in the same input file, the error message appears in the following format: `The Project with id <project id> and version <project version> referenced within <parent resource> with name <parent resource name> and id <parent resource id> and isn't present in the input file.`	
ID null or empty validation	When ID isn't present in the payload, it's generated using the name in the base item details, devoid of all special characters and spaces, and including only characters and numbers. Example: For a base item titled, summer offer 123, the ID generated is `summeroffer123`. ID must not be an empty string for a top-level resource.	ID must not be an empty string for a top-level resource.
Name validation	Name is a mandatory and unique attribute for top-level resources and when null or empty, an error message appears.	Name must be provided for all the top-level resources.
Version validation	The version number must be provided for the entities that are imported.	When the version is either null or empty, a default version 1.0 is created.

Troubleshoot Catalog Export Errors

Here's how you can troubleshoot export job and validation errors:

Error	Error Description	Troubleshooting Tips
Connection failure	The log file generated has the reason for the failure. There can be some errors such as connection failure which means that export job should be tried again after sometime. In case of Launch exception, the error may be related to some specific resource ID not found in the application. That is most common form of error.	In case of connection failure, try the export job after some time.
Application error	Application errors, such as database downtime or some other issues, can't be predicted. Also, sometimes an error may not have occurred at all but the job isn't completed.	Check the scheduled processes for the error reason. It may happen that the job is waiting to be scheduled or the error has occurred because of some Oracle Enterprise Scheduler failures. To check the details of the export job and sub job logs, go to scheduled processes, find the

Error	Error Description	Troubleshooting Tips
		specific job and download it. Verify the job log and the sub job IDs.

7 Publish Catalogs

Publish Catalog Entities

Use this topic to know how you can publish your catalog entities to the spoke systems, such as CRM applications for ordering, billing applications for rating and charging, or provisioning applications for fulfillment.

You can publish catalog definitions using either UI or API using initiative-based publishing.

Before publishing, ensure that you have registered the spoke systems and validated the integration between the applications.

For more information on integration, see CX Industries Framework, Doc ID 2720527.1, on My Oracle Support.

Publishing Process

Here's an overview of the publishing process:

- Register destinations or spoke systems to receive publishing events
- Package catalog entities for publish using initiatives. Optionally, you can use initiatives to seek and acquire approvals for a lifecycle status completion of your catalog entities, for example, design completion.
- Publish initiatives
- Monitor publish jobs

Register destinations or spoke systems to receive publishing events

Register publish destinations before you start the publishing process.

1. Log on to the Launch Experience Cloud Service application as a Catalog Administrator.
2. Go to **Administration > Lifecycle Management**.
3. Click **Save as New Version**. This creates a new LCM version (Pending state) in the editable mode.
4. Select the new version from the combination box if not already selected.
5. Select the Lifecycle Status (LS) state you want to associate with this destination. In this case it would be **Ready to publish**. Click the **Edit** icon on that row.

   > **Note:** Add destinations to the LS where you enabled publishing (Enable Publish = “Y”). Taking the seeded LS as an example, there are two states that have publishing enabled – **Ready to Publish** (for publishing to a test workspace of the spoke system) and **Active** (for publishing to a production workspace of the spoke system).

6. Click **Add Destination**.
7. Enter the following information:
   - Name - <Name of the destination matching with what you will configure in TIC> (CX Industries Framework)
   - Type - <Type of application> - for example, CRM, billing, provisioning and so on
   - Publish Sequence - <The sequence number in which the publishing should happen>
8. Click **Add**.

After you add the destination, complete the following steps:

1. Select the **Destinations** textbox, where the destination you just created would be listed. Select it and save it.
2. On the **Entity Lifecycle Status Configuration** screen, choose the version you just created.
3. Click on the ellipses menu on the right-hand side and click **Activate**.
4. Ensure that the newly created version is in the **Active** state.
5. You've successfully registered a publishing destination in Launch Experience cloud service.
6. Now the same Destination name should be added as a TIC entry to enable the routing rules (see CX Industries Framework, Doc ID 2720527.1, on My Oracle Support)
7. Now you are ready to create a new initiative with artifacts and start publishing to the configured destination.

> **Note:** Ensure that the destination name you configured here (**BuyingTest** in the example above) matches with the Fabric routing rules.

To view sample screenshots of destination configuration for an Active lifecycle status, see *Integrate Launch with Siebel CRM* on My Oracle Support, Document ID 2826909.1.

For sample payloads on how to create a publish destination see REST APIs for Launch on My Oracle Support, *Doc ID 2725039.1* on My Oracle Support.

Package catalog entities for publish using initiatives

For an understanding on catalog entities and how you can collate the entities in an initiative, see the Using Launch Experience guide.

Publish initiatives

When you publish an initiative, the entities such as Catalog, Product Offers, Product Offering Prices and so on are published to spoke systems. You must implement the projectPublishEvent POST operation in the spoke system. To understand the payload and the interface contract, see Oracle Digital Experience for Communications Launch Experience (2725039.1) on My Oracle Support at https://support.oracle.com.

Monitor publish jobs

The publish process is driven by the lifecycle statuses. After you have created a publish job, you can track its status either through API or through the UI.

- To verify status through UI, see the Using Launch Experience guide.
- For publishing acknowledgment REST endpoints, see Oracle Digital Experience for Communications Launch Experience (2725039.1) on My Oracle Support at https://support.oracle.com.

Things to Remember

Do keep the following in mind when preparing to publish:

- You must set up destinations before initiating a publish action. Try a GET on `https://hostname/crmRestApi/resources/11.13.18.05/atcPublishWorkspaceDestinations/` to verify whether destinations are created. If these aren't created, you must create them using POST payload.

  > **Note:** DestinationName should match with the spoke system configuration in CX Industries Framework and the EntryLifecycleStatus must be the initiative lifecycle status from which you would like to publish.

- After you have registered the spoke systems and implemented the projectPublishEvent POST operation in the spoke systems, you must register the spoke system projectPublishEvent endpoint in the routing rules by initiating the Digital Experience for Communications configuration API.

  For more information on setting up routing rules, see CX Industries Framework (2720527.1) on My Oracle Support at https://support.oracle.com. Ensure that the target-preselection-key in the routing rule setup matches with the destination name configured in the first step.
- The payload received in the spoke system for publishing will contain a bunch of Launch Experience API URLs as HREF as well as acknowledgment URLs.
- The spoke system must configure the proper user in Launch Experience with the required roles and privileges, so that this user information can be used to generate OAuth token that will be used when the user will initiate direct Launch API calls to gather more details about the referred objects in Launch Experience.
- After the payload is processed, the spoke system must send the acknowledgment back to Launch Experience about the processing status of the payload using the acknowledgment URL seeded in the payload. The OAuth token must be generated using the same user used in the earlier step.

Related Topics

8 Configure and Extend Launch

Extension Framework

Use this topic to understand the extensibility framework for configuring and extending the Launch application.

Schema-based extension framework is based on the TMF630 extension specification. Application composer is used to build that framework and inheritance is achieved through TMF630 extension patterns. The `@type` and `@basetype` fields in the data model is used to establish the inheritance hierarchy.

Use the Application Composer to create or build the additional extensible fields required to support the extensions. See *About Application Composer*.

To work with schema-based extensions, it's important to understand the three key fields that TMF uses to support and model the extension pattern of entities. All the top-level resources have the three following key fields:

- `@type`: Denotes the resource schema, which defines the resource. You must create and upload the YML file in this name.
- `@basetype`: Denotes the parent resource type.
- `@schemaLocation`: Denotes the URL from which a YML formatted file representing the schema can be read.

Schema-based extension is based on adding new attributes to the existing schema directly. Schema-based extension allows two different ways of extending an resource:

- Adding a simple attribute of type string, number, boolean, date, and so on to the top-level product offer resource.
- Adding a complex attribute of type JSON object or JSON array to the top-level product offer resource.

For an example of the ProductOffering resource represented with specific attributes in ProductOfferingOracle.yml, see the schema-based extension file in Doc ID 2725039.1 on My Oracle Support.

As part of Launch Experience, TMF resources like product offering have been extended to ProductOfferingOracle and similar extensions exist for other resources as well to enable you to do after market extensions in a similar way. An example of an after market extension for a service provider like Vision Inc will be to add a top-level attribute to ProductOfferingOracle schema and name it as ProductOfferingVision.yml.

Customized Roles for Users

You can create or modify child objects that you created as part of the extension framework by adding the role ORA_CRM_EXTN_ROLE to the user.

Extension Types

Use this topic to understand some of the types of extensions that you can do within the Launch application.

You do these extensions using Application Composer and you will need to refer the Application Composer guide for details on some of the aspects covered in this topic. See *About Application Composer*.

Here's a quick start to using Application Composer to create extensions:

- Access Application Composer at runtime by using the Navigator menu, and selecting Application Composer under the Configuration category.
- To make most application changes, you should work in a sandbox. In fact, many functions in Application Composer aren't available until you enter into an active sandbox. You must be an user with Custom Object Administration role to be able to create sandboxes.
- You use sandboxes to make application changes and test them without impacting other users in the environment. Wherever possible, make changes to the application in a sandbox rather than making direct changes in the mainline environment.
- Make object or schema changes in Application Composer and publish changes.

  > **Note:** After the extensions are published, they can't be reverted.

- Make changes to schema files and upload them to UCM via API.
- Start using the extended attributes via the APIs.

Simple Attribute Extension

To understand how you can implement a simple attribute extension, let's take the example of a service provider like Vision Inc. To perform an extension, Vision Inc would add a top-level attribute to the ProductOfferingOracle schema and name it ProductOfferingVision.yml.

1. Go to Application Composer.
2. Select **CRM Cloud** from the Application drop-down list.
3. Click **Standard Objects** on the Objects Explorer.

   All the standard objects shipped by the application are displayed including that of other Oracle Fusion applications.

   > **Note:** There are two sections, Custom and Standard. Standard fields are the base fields based on TMF resource specifications and there's no access to edit the standard fields. See the Resource Details table for a list of the TMF 620 Resources, Oracle Fusion applications standard objects that support extension along with their YML schema names. For the example, Product Offering standard object is used.

4. Click **Product Offering > Fields**.
5. On the Fields page, Custom section, click **Create**.
6. On the Select Field Type dialog box, select a corresponding type, for example, **Text** and click **OK**.

   The data type supported in Application Composer includes strings, numbers, boolean, date, datetime, and choice list (`enums`). In this example, the simple attribute VisionIncName is a text field.
7. On the Create Text Field dialog box, fill in the following:
   a. Display Name: Specify the Display name as Vision Inc Name. The display name is the view name of the attribute in Application Composer.
   b. Display Label: Select a Display Label, for example, Vision Inc Name. The Display Label field retains the name of the attribute used for the Application Composer object.
   c. API Name: It's auto generated with respect to the DisplayName, by trimming the white space, for example, VisionIncName. This name must be used in the service payload. If you change it, the same must be used in service payload.
   d. Description: Provide a description of the attribute
   e. Constraints:

        i. Required: Selecting this makes the attribute mandatory with the service payload. It's suggested to not enable this option for a custom attribute because this affects the base object. If you prefer to make an attribute or object as required, you can restrict in schema.
        ii. Updatable: This is checked by default. Disabling this will fail the update operation.
        iii. Searchable: This is related to Application Composer REST service and isn't used in this scenario.
        iv. Indexed: This is related to Application Composer REST service and isn't used in this scenario.
        v. Minimum Length: A text field specific constraint used to define the minimum length supported. By default the value is set to 0.
        vi. Maximum Length: A text field specific constraint used to define the maximum length supported. The default length is 80.
    f. Default Value: Fixed Value: You can define a static value for the payload, for a boolean field.
8. Add lookups or `enums` for the custom attribute:
    a. Click **Fields** within the corresponding standard object.
    b. On the Fields page, Custom section, click **Create**.
    c. On the Select Field Type dialog box, select a corresponding type. You must select **Choice List** and click **OK**.
    d. Enter the following information:
        i. Display Name: This field is mandatory. The display name is visible for a user in Application Composer.
        ii. API Name: This gets generated automatically based on the display name and is used in the payload.
        iii. Required: Selecting this makes the attribute mandatory with the service payload. It's suggested to not enable this option for a custom attribute because this affects the base object. If you prefer to make an attribute or object as required, you can restrict in schema.
        iv. Updatable: This is checked by default. Disabling this will fail the update operation.
        v. Searchable: This is related to Application Composer REST service and isn't used in this scenario.
    e. Click **Add** in the Lookup Type section and add the following information, if the extended field is a lookup type:

        > **Note:** You need not perform the following steps if it's a string with a free form text.

        i. Meaning: This field is mandatory as it serves as the display name.
        ii. Lookup type: Code to identify lookup. Unique code to identify the list.
        iii. Description: Add description about the lookup.
        iv. To add values to lookup, click **Actions> Create**.
        v. Meaning: This field is mandatory as it serves as the display name.
        vi. Lookup type: Code to identify lookup value.
        vii. Description: Add description about the lookup.
        viii. Enabled: Select to either enable or disable a particular value usage.
        ix. Display Sequence: Specify the order of display.
9. Click **Save** and ensure that the new field, VisionIncName, is added to the Custom Field section.
10. Go to sandbox and publish the sandbox.

    > **Note:** You can perform testing only after publishing the sandbox.

11. After the sandbox is published, create the schema file with the simple attribute that you just created and refer to the ProductOfferingOracle schema.

> **Note:** The extension will be available in the instance at this point.

```
title: ProductOfferingVision
type: object
description: 'It's designed to model flexible interface definition in order to provide maximal reuse for
 existing rating algorithms.'
allOf:
 - $ref: 'ProductOfferingOracle.yml'
 - properties:
 VisionIncName:
 type: string
 description: Name of partner
```

12. Save it as ProductOfferingVision.yml and upload to custom folder using schema endpoint.
13. After the schema is uploaded, use the REST API endpoints to POST and GET data. Here's a sample payload and the steps to go about testing:
    a. Publish the sandbox changes.
    b. Upload the custom schema created with extension attributes to custom folder using schema upload endpoint.
    c. Use the resource endpoint to trigger extension. For example, https://<hostname>/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/productOffering.

```
{
 "id": "SUP_PS_WS_ROAM_105",
 "name": "Wireless Roaming PS 105",
 "description": "Product Specification for Wireless Roaming",
 "version": "1.0",
 "lifecycleStatus": "In design",
 "@type": "ProductOfferingVision",
 "@baseType": "ProductOfferingOracle",
 "project": {
 "name": "P_16_9_2020_11_55_15",
 "id": "P_16_9_2020_11_55_15"
 },
 "validFor": {
 "startDateTime": "2020-09-14T00:00:00.000Z"
 },
 "VisionIncName": "Amazefit-v"
}
```

Complex Attribute Extension: Object and Array

You can perform an aftermarket extension using complex type extension by adding array or object to the existing resources. Application Composer supports only child object as array type, which is the default. There's no place in Application Composer where you can specify the type of child that you want to extend, for example, array or object. This can be achieved with custom schema. During schema creation you must restrict the type of child to be created in Application Composer as array or object. Here's how you can do it.

Object Type Extension

For example, let's introduce a complex object called PartnerDetailsOffering within the product offering resource. The PartnerDetails is an object which contains two text fields PartnerName and PartnerCode. You must begin by enabling Application Composer through sandboxes. You must be an user with Custom Object Administration role to be able to create sandboxes. For more information on how to create and activate sandboxes, see *Create and Activate Sandboxes*.

1. Go to Application Composer.
2. Click **Standard Objects**.
3. Click **Product Offering**.
4. On the ProductOffering: Overview page, click **Create Child Object**.

5. On the Create Child Object dialog box, enter the following information:
    1. Display Name: This name is used to display the object in Application Composer. As naming standard display name carries proper white spaces, the object name is displayed as Partner Details. Based on the display name, the rest of the fields get auto populated.
    2. API Name: This name used in service payload. The API name gets generated based on the display name, by trimming the white space, and is displayed as PartnerDetails. You can't modify the API name after creating it. Also, you must provide the same name as object name in the payload.
6. Click **Save** to create the child object. The same now appears within the ProductOffering object in the Standard Objects section.
7. Expand the Partner Details child object and click **Fields**. There will be two sections similar to simple attribute, which are Custom and Standard.
8. To create PartnerName as well as PartnerCode, click **Create** in the Custom section.
9. On the Select Field Type dialog box, select a corresponding type, for this example, **Text** and click **OK**.
10. On the Create Text Field dialog box, fill in the following:
    1. Display Name: Specify the display name. All other fields will be auto generated.
    2. Display Label: Select a Display Label. This field retains the name of the attribute used for the Application Composer object. However, it isn't used in the payload or on the UI.
    3. API Name: It's auto generated with respect to the DisplayName, by trimming the white space. This name must be used in the service payload. If you change it, the same must be used in service payload.
    4. Description: Provide a description of the attribute
    5. Constraints:
        1. Required: Selecting this makes the attribute mandatory with the service payload. It's suggested to not enable this option for a custom attribute because this affects the base object. If you prefer to make an attribute or object as required, you can restrict in schema.
        2. Updatable: This is checked by default. Disabling this will fail the update operation.
        3. Searchable: This is related to Application Composer REST service and isn't used in this scenario.
        4. Indexed: This is related to Application Composer REST service and isn't used in this scenario.
        5. Minimum Length: A text field specific constraint used to define the minimum length supported. By default the value is set to 0.
        6. Maximum Length: A text field specific constraint used to define the maximum length supported. The default length is 80.
    6. Default Value: Fixed Value: You can define a static value for the payload, for a boolean field.
    7. Add lookups or `enums` for the custom attribute:
        1. Click **Fields** within the corresponding standard object.
        2. On the Fields page, Custom section, click **Create**.
        3. On the Select Field Type dialog box, select a corresponding type. You must select **Choice List** and click **OK**.
        4. Enter the following information:
            1. Display Name: This field is mandatory. The display name is visible for a user in Application Composer.
            2. API Name: This gets generated automatically based on the display name and is used in the payload.
            3. Required: Selecting this makes the attribute mandatory with the service payload. It's suggested to not enable this option for a custom attribute because this affects the base object. If you prefer to make an attribute or object as required, you can restrict in schema.

        d. Updatable: This is checked by default. Disabling this will fail the update operation.
        e. Searchable: This is related to Application Composer REST service and isn't used in this scenario.
    v. Click **Add** in the Lookup Type section and add the following information:
        a. Meaning: This field is mandatory as it serves as the display name.
        b. Lookup type: Code to identify lookup. Unique code to identify the list.
        c. Description: Add description about the lookup.
        d. To add values to lookup, click **Actions> Create**.
        e. Meaning: This field is mandatory as it serves as the display name.
        f. Lookup type: Code to identify lookup value.
        g. Description: Add description about the lookup.
        h. Enabled: Select to either enable or disable a particular value usage.
        i. Display Sequence: Specify the order of display.
h. Click **Save** and verify the changes.
i. Go to sandbox and publish the sandbox.

Here's the custom schema for type object.

```
title: PartnerDetailsOffering
type: object
description: 'It's designed to model flexible interface definition in order to provide maximal reuse for
 existing rating algorithms.'
allOf:
 - $ref: 'ProductOfferingOracle.yml'
 - properties:
 PartnerDetails:
 type: object
 description: Partner details Reference
 properties:
 PartnerName:
 type: string
 description: partner name details
 PartnerCode:
 type: string
 description: partner code details
```

The type of child created is referred to as object so the extension will only support the format of objects. The schema is saved as PartnerDetailsOffering.yml and uploaded to the custom folder using schema endpoint as described in Upload schema section. For more information on uploading schema, see the section in this topic on uploading schema via Launch schema endpoint.

After the schema is uploaded, use the REST API endpoints to POST and GET data. Here's a sample payload and the steps to go about testing:

1. Publish the sandbox changes and verify the changes.
2. Upload the custom schema created with extension attributes to custom folder using schema upload endpoint.
3. Use the resource endpoint to trigger extension. For example, https://<hostname>/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/productOffering.

To test the scenario, the following payload can be used, with @type as PartnerDetailsOffering to trigger the extension.

```
{
 "id": "SUP_PS_WS_ROAM_203",
 "name": "Partner Details Spec 203",
 "description": "Product Specification for Wireless Roaming",
 "version": "1.0",
 "lifecycleStatus": "In design",
 "@type": "PartnerDetailsOffering",
 "@baseType": "ProductOfferingOracle",
 "project": {
 "name": "P_16_9_2020_11_55_15",
 "id": "P_16_9_2020_11_55_15"
 },
 "validFor": {
 "startDateTime": "2020-09-14T00:00:00.000Z"
 },
 "PartnerDetails": {
 "PartnerName": "Airtel",
 "PartnerCode": "AT-01"
 }
}
```

Array Type Extension

The workflow for an array type extension is similar to that of a custom object type. Here the child object is treated as an array type. For the example, an array ConsumerList containing fields, ConsumerName and ConsumerCode, are used. Application Composer can be enabled only through sandboxes. You must be an user with Custom Object Administration role to be able to create sandboxes. For more information on how to create and activate sandboxes, see *Create and Activate Sandboxes*.

1. Go to Application Composer.
2. Click **Standard Objects**.
3. Click **Product Offering**.
4. On the ProductOffering: Overview page, click **Create Child Object**.
5. On the Create Child Object dialog box, enter the following information:
   a. Display Name: This name is used to display the object in Application Composer. As naming standard display name carries proper white spaces, the object name is displayed as Consumer List. Based on the display name, the rest of the fields get auto populated.
   b. API Name: This name used in service payload. The API name gets generated based on the display name, by trimming the white space, and is displayed as ConsumerList. You can't modify the API name after creating it. Also, you must provide the same name as object name in the payload.
6. Click **Save**. The same now appears within the ProductOffering object in the Standard Objects section.
7. Expand the Consumer List child object and click **Fields**. There will be two sections similar to simple attribute, which are Custom and Standard.
8. To create ConsumerName as well as ConsumerCode, click **Create** in the Custom section.
9. On the Select Field Type dialog box, select a corresponding type, and click **OK**.
10. On the Create Field dialog box, fill in the following:
    a. Display Label: Select a Display Label. This field retains the name of the attribute used for the Application Composer object. However, it isn't used in the payload or on the UI.
    b. API Name: It's auto generated with respect to the DisplayName, by trimming the white space. This name must be used in the service payload. If you change it, the same must be used in service payload.
    c. Description: Provide a description of the attribute
    d. Constraints:
       i. Required: Selecting this makes the attribute mandatory with the service payload. It's suggested to not enable this option for a custom attribute because this affects the base object. If you prefer to make an attribute or object as required, you can restrict in schema.
       ii. Updatable: This is checked by default. Disabling this will fail the update operation.
       iii. Searchable: This is related to Application Composer REST service and isn't used in this scenario.
       iv. Indexed: This is related to Application Composer REST service and isn't used in this scenario.

v. Minimum Length: A text field specific constraint used to define the minimum length supported. By default the value is set to 0.
vi. Maximum Length: A text field specific constraint used to define the maximum length supported. The default length is 80.

e. Default Value: Fixed Value: You can define a static value for the payload, for a boolean field.
f. Add lookups or `enums` for the custom attribute:

   i. Click **Fields** within the corresponding standard object.
   ii. On the Fields page, Custom section, click **Create**.
   iii. On the Select Field Type dialog box, select a corresponding type. You must select **Choice List** and click **OK**.
   iv. Enter the following information:

      a. Display Name: This field is mandatory. The display name is visible for a user in Application Composer.
      b. API Name: This gets generated automatically based on the display name and is used in the payload.
      c. Required: Selecting this makes the attribute mandatory with the service payload. It's suggested to not enable this option for a custom attribute because this affects the base object. If you prefer to make an attribute or object as required, you can restrict in schema.
      d. Updatable: This is checked by default. Disabling this will fail the update operation.
      e. Searchable: This is related to Application Composer REST service and isn't used in this scenario.

   v. Click **Add** in the Lookup Type section and add the following information:

      a. Meaning: This field is mandatory as it serves as the display name.
      b. Lookup type: Code to identify lookup. Unique code to identify the list.
      c. Description: Add description about the lookup.
      d. To add values to lookup, click **Actions> Create**.
      e. Meaning: This field is mandatory as it serves as the display name.
      f. Lookup type: Code to identify lookup value.
      g. Description: Add description about the lookup.
      h. Enabled: Select to either enable or disable a particular value usage.
      i. Display Sequence: Specify the order of display.

g. Click **Save** and verify the changes.
h. Go to sandbox and publish the sandbox.

The schema is modified as type array.

```
title: ConsumerListOffer
type: object
description: 'It's designed to model flexible interface definition in order to provide maximal reuse for
 existing rating algorithms.'
allOf:
 - $ref: 'ProductOfferingOracle.yml'
 - properties:
 ConsumerList:
 type: array
 description: Consumer List Reference
 items:
 type: object
 description: Consumer List Reference
 properties:
 ConsumerName:
 type: string
 description: Consumer name details
 ConsumerCode:
 type: string
 description: Consumer code details
```

The schema is saved as ConsumerListOffer.yml and uploaded to the custom folder using schema endpoint as described in Upload schema section. For more information on uploading schema, see the section in this topic on uploading schema via Launch schema endpoint.

After the schema is uploaded, use the REST API endpoints to POST and GET data. Here's a sample payload and the steps to go about testing:

1. Publish the sandbox and verify the changes.
2. Upload the custom schema created with extension attributes to custom folder using schema upload endpoint.
3. Use the resource endpoint to trigger extension. For example, https://<hostname>/crmRestApi/atcProductCatalog/11.13.18.05/tmf-api/productCatalogManagement/v4/productOffering.

To test the scenario, the following payload can be used, with @type as ConsumerListOffer to trigger the extension.

```
{
 "id": "SUP_PS_WS_ROAM_301",
 "name": "Consumer List Spec 302",
 "description": "Product Specification for Wireless Roaming",
 "version": "1.0",
 "lifecycleStatus": "In design",
 "@type": "ConsumerListOffer",
 "@baseType": "ProductOfferingOracle",
 "project": {
 "name": "P_16_9_2020_11_55_15",
 "id": "P_16_9_2020_11_55_15"
 },
 "validFor": {
 "startDateTime": "2020-09-14T00:00:00.000Z"
 },
 "ConsumerList": [
 {
 "ConsumerName": "Consumer 1",
 "ConsumerCode": "CS-01"
 },
 {
 "ConsumerName": "Consumer 2",
 "ConsumerCode": "CS-02"
 }
 ]
}
```

Note:

- You can't undo the changes for simple attribute or complex attribute objects after publishing the sandbox. Once it's published the changes will remain. So, creating a simple extension attribute with Required as enabled isn't recommended. To make an attribute as required, you must modify the schema and mark it as required within the schema.
- Application composer can't handle complex extension types such as object or array specifically. Creating child objects and adding custom fields are common to both the scenarios. You can restrict the type of extension in schema only as explained earlier.

Migration of Extensions

Any implementation of Oracle Applications Cloud usually requires migrating extensions from one environment to another at various points in the subscription lifecycle. It's recommended to start with a test instance, test the changes, and only after validation, apply the changes to the production instance.

The extension configured in one instance can be moved to another either by manually configuring or by using the migration tool for Application Composer related changes.

For schema changes, you must manually move the changes after migrating Application Composer changes. Custom schema can be uploaded to the custom folder using schema endpoint as described in the following section.

Upload Schema via Launch Experience Schema Endpoint

Use the Launch Experience schema endpoint to upload new or updated YML files. Here's how you can go about it:

1. Go to endpoint: https://<hostname>/crmRestApi/atcProductCatalog/11.13.18.05/v1/schema.
2. Set Method to POST (to add new schema) or PUT (to overwrite an existing schema).
3. Request Body: Set mimeType as multipart or form-data.
4. Add the following rows inside the form:

Key	Value
path	custom
primaryFile	Select the file to be uploaded

5. Send the request.
6. Validate to ensure that the extension works.

Resource Details

Use this topic to understand the base and the extended resources shipped with the Launch application. You will require the resource details to be able to extend these resources using schema-based extension.

This table contains the resources with their corresponding YML names.

Catalog Resources	Oracle Shipped Extensions	YML Schema Name	Application Composer Object Name
Product Offer	ProductOfferingOracle	ProductOfferingOracle.yml	Launch Product Offerings
Product Offering Price	• ProductOfferingPriceOracle • ProductOfferPriceAllowanceOr • ProductOfferPriceOverageOrac • ProductOfferPricePlanOracle • ProductOfferPriceAlterationOra	• ProductOfferingPriceOracle.ym • ProductOfferPriceAllowanceOr • ProductOfferPriceOverageOrac • ProductOfferPricePlanOracle.y • ProductOfferPriceAlterationOra	Launch Product Offering Prices

Catalog Resources	Oracle Shipped Extensions	YML Schema Name	Application Composer Object Name
Product Specification	ProductSpecificationOracle	ProductSpecificationOracle.yml	Launch Product Specifications
Pricing Logic Algorithm	PricingLogicAlgorithmOracle	PricingLogicAlgorithmOracle.yml	Launch Pricing Logic Algorithms
PLA Specification	- PricingLogicAlgorithmSpecOra - UsagePLASpecOracle - OneTimePLASpecOracle - RecurringPLASpecOracle	- PricingLogicAlgorithmSpecOra - UsagePLASpecOracle.yml - OneTimePLASpecOracle.yml - RecurringPLASpecOracle.yml	Launch Pricing Logic Algorithm Specifications
Product Line	ProductLineOracle	ProductLineOracle.yml	Launch Product Lines
Project	ProjectOracle	ProjectOracle.yml	Launch Initiatives
Catalog	CatalogOracle	CatalogOracle.yml	Launch Catalogs
Category	CategoryOracle	CategoryOracle.yml	Launch Categories
Customer Profile Specification	CustomerProfileSpecificationOracle	CustomerProfileSpecificationOracle.	Launch Customer Profile Specifications
Service Specification	ServiceSpecificationOracle	ServiceSpecificationOracle.yml	Launch Service Specifications
Tax Service Providers	TaxServiceProviderOracle	TaxServiceProviderOracle.yml	Launch Tax Service Providers
Usage Specification	UsageSpecificationOracle	UsageSpecificationOracle.yml	Launch Usage Specifications
Balance Element	BalanceElementOracle	BalanceElementOracle.yml	Launch Balance Elements
Promotion	PromotionOracle	PromotionOracle.yml	Launch Promotions
Custom Profile Specification	CustomProfileSpecificationOracle	CustomProfileSpecificationOracle.yr	Launch Custom Profile Specifications
Price List	PricelistOracle	PricelistOracle.yml	Launch Price Lists
Price Constraints	PricelistConstraintOracle	PricelistConstraintOracle.yml	Launch Price Constraints
Lifecycle Configuration	LifecycleConfigurationOracle	LifecycleConfigurationOracle.yml	Launch Lifecycle Configurations

Configure Lifecycle Status

Use this topic to understand how you can configure the lifecycle status and where you can find more information regarding the configuration.

The lifecycle configuration resource represents a set of lifecycle configurations and lifecycle states. Lifecycle of all resources transition based on the lifecycle configuration associated to the project to which a resource is added.

You can extend lifecycle configurations by adding more lifecycle states and defining the characteristics for a state by creating a new version of the lifecycle configuration. You can add custom states only between In design and Launched.

To understand how you can add a new version of lifecycle status, see *Configure a New Lifecycle Status*.

Extend Lookup Values

Use this topic to understand how you can add lookup values using the extensibility framework.

You can create additional values or extend the default value sets for lookups and make modifications as per your business requirements. Review the existing enumerations, if any, for the lookup value that you're changing and if you need to have the validation at the API layer, then it must be added to the enumerations.

Let's take the example of setting up the eligibility parameters and see how you can modify values for a lookup.

1. Go to **Navigator > My Enterprise > Setup and Maintenance**.
2. In Setup and Maintenance, click **Tasks > Search** and enter **Manage Standard Lookups** in the Search field.
3. On the Manage Standard Lookups page, search for the lookup either by name or code. For our example, search for the ORA_ATC_PARMTR to edit the eligibility parameter lookup. The ORA_ATC_PARMTR lookup type now appears in the Search Results section.

   If you're not sure about the code, you can search with ORA_ATC_% in the Lookup Type section, which will show you all the LOVs in the application.
4. In the ORA_ATC_PARMTR: Lookup Codes section, update the LOVs as required. You can modify, add, or delete values.
5. Click **Save** after you're done with your changes.
6. Click **Save and Close**.

For more information about lookups, see *Overview of Lookups*.

Extend Finance Plan

Use this topic to know how you can add lookup values to the existing finance plan.

The finance plan feature enables you to factor in interest based installment plans with amortization details (in addition to the current 0% interest based installment plans). You can create additional lookup values or extend the amortization attribute sets and make changes as per your business requirements. The extension to these attributes is currently supported only through the API layer.

Seeded Attributes

The following table shows the seeded attributes shipped with the Launch application.

Amortization Characteristics	Characteristic Values
Interest Type	Simple Compound Factor None
Pattern Type	Simple Range
Duration	Months Years
Interest Details Sub-characteristics: • Range From • Range To • Interest	
Entitlement Templates	

You can add a new amortization characteristic or add new sub characteristics within the interest details. To add a new finance plan, you need to execute a PUT call on the REST API endpoint on `crmRestApi/atcProductCatalog/11.13.18.05/productCatalogReferenceManagement/v1/customProfileSpecification/{id}` with the profileType `FINANCE_PLAN`.

To access the finance plan template and to know about other related REST API details, see REST APIs for Launch Experience on My Oracle Support, *Doc ID 2725039.1*

Add a Simple Characteristic

You can add a simple characteristic which takes free form text as input.

```
{
 "name": "COMPOUNDING TERM",
 "description": "If the interest type is Compound ,if it is compounded annually,half yearly or quarterly",
 "valueType": "STRING",
 "active": true
 }
```

Add Sub characteristics within Interest Details

To extend the interest details characteristic, specify the relation of the characteristic to the interest details, along with the following values.

```
{
```

```
"name": "InterestConcession",
"customProfileSpecCharValue": [
{
"valueTo": "1",
"valueType": "DECIMAL",
"isDefault": false
},
{
"valueTo": "2",
"valueType": "DECIMAL",
"isDefault": false
},
{
"valueTo": "3",
"valueType": "DECIMAL",
"isDefault": false
}
],
"customProfileSpecCharRel": [
{
"charSpecSeq": 1,
"id": "GlobalFinancePlan",
"relationshipType": "PARENT",
"name": "interestDetails"
}
]
}


The Payload to Extend the Finance Plan can be found in the < Replace with Payload page URL to Sample Payload
 CustomProfileSpec>
```

Extended Attributes in Dynamic Forms

You can customize dynamic forms to show extended attributes for offers, product specification, and catalog.

You can edit the basic details form of these objects using VB Studio. Refer the *VB Studio guide* to understand how to make the changes.

You would need to do the following steps for Launch:

1. From the Settings and Actions menu in Visual Builder, click **Edit Pages** to open the page in the Designer in VB Studio.
2. Choose the form layout you need to edit. You may either duplicate the layout and make changes or create a new layout.
3. Add or remove fields from the new layout as needed. For the new fields, make sure that you create a new field and template under the Field as well as Template section. Also ensure that the field is correctly mapped to the template under the fieldTemplateMap section in the layout JSON. Here's a sample input text template for a custom Partner ID field for product offering:

   ```
   <template id="partnerIdTemplate">
    <oj-input-text value="{{ $componentContext.pageVariables.productOfferingPayload.PartnerId }}" label-
   hint="[[ $metadata.labelHint ]]"
    readonly="[[ $componentContext.pageVariables.offerFlow === $layout.functions.constants.data_mode.view
    || $componentContext.pageVariables.offerFlow === $layout.functions.constants.data_mode.revise]]"></oj-
   input-text>
   </template>
   ```

   You may refer the read-only definition of other fields in the layout template to use the correct variables.

For the input value to be included in the payload, you need to use two-way binding using curly braces to the correct variable as shown in the sample. In the sample, `$componentContext.pageVariables.productOfferingPayload` is the payload variable used for product offering. Similarly use `$componentContext.pageVariables.productSpecificationPayload` for Product Specification and `$componentContext.pageVariables.catalogPayload` for Catalog.

After configuring the new layout, edit the rule set to show your layout based on `_attype`. A sample rule set definition for Product Offering, Product Specification and Catalog is as follows:

-
  ```
  $componentContext.pageVariables.productOfferingPayload._attype ===
   'PartnerOffer'

  ? 'partnerLayout' : 'default'
  ```

-
  ```
  $componentContext.pageVariables.productSpecificationPayload['@type'] ===
   'TestProductSpecification'

  ||
   $componentContext.pageVariables.getProductSpecificationByIdResponse['_attype'] ===
   'TestProductSpecification'

  ? 'customLayout' : 'default'
  ```

-
  ```
  $componentContext.pageVariables.catalogPayload['@type'] ===
   'TestCatalog'

  ? 'partnerCatalog' : 'default'
  ```

Configure Layouts for Product Specifications

Use this topic to determine how you can create a layout file to display the extended product specification characteristics. These characteristics are displayed in the attributes train stop of the pages for creating or editing offers. A product specification can

Prerequisites

Here are some of the prerequisites:

- The layout file name must be of the format: <Product Specification Id> -layout.json. For example, 5GSupremo-layout.json.
- The layout file must be uploaded to the CatalogManagement/schema/Layouts directory in UCM. For steps on how to upload the layout file, refer *Extension Types*.
- The layout file must have the following schema:

```
{
 "$schema": "http://json-schema.org/schema#",
 "id": "http://example.com/schemas/jet/npi-dynamic-layout-schema.json",
 "title": "NPIDynamicLayout",
 "description": "Describes the model for the NPI Dynamic Layout",
 "definitions": {
 "propertyLayout": {
 "description": "Defines a layout structure for a property",
 "type": "object",
 "properties": {
 "labelHint": {
 "description": "Defines label for the property",
```

```
"type": "string"
},
"class": {
"description": "Class to be applied to this property. Used for overriding input components",
"type": "string"
}
},
"additionalProperties": false
},
"displayProperties": {
"description": "Defines structure of displayProperties used by formLayout",
"type": "array",
"items": {
"anyOf": [{
"type": "string"
}, {
"type": "object",
"patternProperties": {
"^[\\$A-Z_a-z][\\$A-Z_a-z0-9/-]*$": {
"$ref": "#/definitions/propertyLayout"
}
}
}]
}
},
"formLayout": {
"properties": {
"labelHint": {
"type": "string"
},
"displayProperties": {
"$ref": "#/definitions/displayProperties"
}
},
"additionalProperties": false
},
"singleLayout": {
"description": "Definition of the single layout structure",
"type": "object",
"properties": {
"description": {
"type": "string"
},
"layoutType": {
"type": "string"
},
"layout": {
"$ref": "#/definitions/formLayout"
}
},
"additionalProperties": false
}
},
"type": "object",
"properties": {
"layouts": {
"description": "The layouts definition",
"patternProperties": {
"^[\\$A-Z_a-z][\\$A-Z_a-z0-9/-]*$": {
"$ref": "#/definitions/singleLayout"
}
},
"additionalProperties": false
}
},
"additionalProperties": false
```

```
    }
```

How To Go About It

After you have set up your prerequisites, set up the UI layouts in the following way.

Add and Order Fields in a Layout

To make a field visible on the UI, you must define it in layout.json. The order of the fields in the UI is the order of the fields in layout.json. Here's an example.

```
    {
     "layouts": {
     "phone-form": {
     "description": "layout for phone form with grouping support",
     "layoutType": "form",
     "layouts": {
     "apple": {
     "layoutType": "form",
     "layout": {
     "description": "form layout for apple iPhone",
     "displayProperties": ["storageSpace", "chipName", "batteryCapacity", "operatingSystem"]
     }
     }
     }
     }
     }
    }
```

Create Groups in a Layout

Here's an example that shows how you can create groups.

```
    {
     "layouts": {
     "phone-form": {
     "description": "layout for phone form with grouping support",
     "layoutType": "form",
     "layouts": {
     "apple": {
     "layoutType": "form",
     "layout": {
     "description": "form layout for apple iPhone",
     "displayProperties": [{
     "physical": {
     "description": "layout for physical group",
     "layoutType": "form",
     "layout": {
     "labelHint": "Physical",
     "displayProperties": ["widthValue", "heightValue", "weightValue", "dateOfManufacture", "dateOfRelease",
     "detailedDescription"]
     }
     }
     }, {
     "technical": {
     "description": "layout for technical group",
     "layoutType": "form",
     "layout": {
     "labelHint": "Technical",
     "displayProperties": ["storageSpace", "chipName", "batteryCapacity", "operatingSystem"]
     }
     }
     }, {
     "logistics": {
```

```
 "description": "layout for logistics group",
 "layoutType": "form",
 "layout": {
 "labelHint": "Logistics",
 "displayProperties": ["shippingCarriers", "vendorPartNum", "shippingTime"]
 }
 }
 }]
 }
 }
 }
 }
 }
}
```

Override Labels

You must use the labelHint property to override UI labels. Here's an example of shippingMethod that shows how to override labels.

```
{
 "layouts": {
 "phone-form": {
 "description": "layout for phone form with grouping support",
 "layoutType": "form",
 "layouts": {
 "apple": {
 "layoutType": "form",
 "layout": {
 "description": "form layout for apple iPhone",
 "displayProperties": [{
 "physical": {
 "description": "layout for physical group",
 "layoutType": "form",
 "layout": {
 "labelHint": "Physical",
 "displayProperties": ["widthValue", "heightValue", "weightValue", "dateOfManufacture", "dateOfRelease",
 "detailedDescription"]
 }
 }
 }, {
 "technical": {
 "description": "layout for technical group",
 "layoutType": "form",
 "layout": {
 "labelHint": "Technical",
 "displayProperties": ["storageSpace", "chipName", "batteryCapacity", "operatingSystem"]
 }
 }
 }, {
 "logistics": {
 "description": "layout for logistics group",
 "layoutType": "form",
 "layout": {
 "labelHint": "Logistics",
 "displayProperties": ["shippingCarriers", {"shippingMethod":{"labelHint": "Available Shipping Methods"}},
 "shippingTime"]
 }
 }
 }]
 }
 }
 }
 }
 }
```

```
    }
```

Override UI Controls

You must use the class property to override UI controls. Here's an example of shippingRequiredFlag that shows how to override the input component for a field.

```
    {
     "layouts": {
     "phone-form": {
     "description": "layout for phone form with grouping support",
     "layoutType": "form",
     "layouts": {
     "apple": {
     "layoutType": "form",
     "layout": {
     "description": "form layout for apple iPhone",
     "displayProperties": [{
     "physical": {
     "description": "layout for physical group",
     "layoutType": "form",
     "layout": {
     "labelHint": "Physical",
     "displayProperties": ["widthValue", "heightValue", "weightValue", "dateOfManufacture", "dateOfRelease",
     "detailedDescription"]
     }
     }
     }, {
     "technical": {
     "description": "layout for technical group",
     "layoutType": "form",
     "layout": {
     "labelHint": "Technical",
     "displayProperties": ["storageSpace", "chipName", "batteryCapacity", "operatingSystem"]
     }
     }
     }, {
     "logistics": {
     "description": "layout for logistics group",
     "layoutType": "form",
     "layout": {
     "labelHint": "Logistics",
     "displayProperties": ["shippingCarriers", "vendorPartNum", {"shippingRequiredFlag":{"class":"boolean-
    radioset-yn"}}, "shippingTime"]
     }
     }
     }]
     }
     }
     }
     }
     }
    }
```

Override Field Controls

You must use the following set of values.

Boolean

What you use	How it's rendered
boolean-checkbox	As a check box. This is the default.

What you use	How it's rendered
boolean-radioset-yn	As a radio set with a Yes or No option.
boolean-radioset-tf	As a radio set with a True or False option.
boolean-dropdown-yn	As a drop-down list with a Yes or No option.
boolean-dropdown-tf	As a drop-down list with a True or False option.

LOV

What you use	How it's rendered
lov-dropdown	As a drop-down list. This is the default.
lov-radioset	As a radio set.

9 Integrate Launch with Other Applications

Entity Profile

When you integrate Launch with third-party target systems, you need to respect the boundaries of the source and target systems. Entity profiles help you to create these boundaries between source and target systems.

Launch can consume catalogs from other applications as a part of its coexistence strategy. For example, Launch can consume and master any third-party catalogs.

Entity profiles are constructs that incorporate a set of rules used by Launch to ensure the following:

- Migrate entities smoothly from runtime systems based on their features.
- Apply restrictions to protect the shape of the migration from modification in Launch.
- Publish changes or revisions made to the migrated entities in Launch back to and without breaking, the target runtime application.

A migrated resource must have a resource-specific entity profile associated with every instance of the resource, to respect the source ecosystem.

Integrate Launch with Buying Experience

Take advantage of the prebuilt integration between Launch Experience and Buying Experience. Publish the design-time constructs of catalog definitions to the runtime ordering applications.

You can publish every entity modeled in Launch Experience to the Buying Experience using the CX Industry Framework. The entities include product offers, pricing, product lines, terms, rules, catalog, categories, product and service specifications, and others.

Setting Up the Integration

To set this integration up, ensure that you have registered the Buying Experience and validated the integration between the applications. For more information, see:

- *Implement CX Industries Framework* on My Oracle Support, Document ID 2720527.1.
- *Implementing Buying Experience* on Oracle Help Center.

Integrate Launch with Siebel CRM

Take advantage of the prebuilt integration between Launch and Siebel CRM to centrally manage product and services portfolio across your ecosystem. Launch is the master for catalog definitions.

Siebel customers can innovate faster by centrally managing their Product Portfolio for their entire eco-system, including Siebel CRM.

The prebuilt integration between the two applications allows migration of catalog definitions from Siebel and master them in Launch.

Versions Supported

- Launch version 22.10 or later
- Siebel CRM version 22.11 or later

What Can You Migrate or Publish?

The following table lists the entities you can migrate or publish. The table also lists the predefined mappings available to you in the JSON file.

Entity Name in Siebel CRM	Entity Name in Launch	What you can Sync
Catalog	Catalog	• Definition • Category Association
Category	Category	• Definition • Subcategories • Product Association
Product Class	Product Specification	• Definition • Parent-Child Relationships • Attributes • Smart Part Number (user defined)
Products	Simple Offer, Bundle Offer	• Definitions • Category/Class/Product Line association • Prices and Adjustments • Volume Discounts • Compatibility and Eligibility Rules • Recommendation and Migration Rules • Constraint Rules • Attribute Adjustments
Product Line	Product Line	• Definition • Product association • Compatibility and Eligibility Rules
Price list	Price list	Definition and its associations
Promotion	Package	• Definitions

Entity Name in Siebel CRM	Entity Name in Launch	What you can Sync
		- Category/Class/Product Line association - Components along with Aggregates and Option Class - Compatibility and Eligibility Rules - Recommendation and Migration Rules - Commitment Terms
Aggregate Discounts and Sequences	Promotion	Definition and its associations
Entitlement Templates	Entitlement Templates	Definition and its associations

Set Up the Integration

To set this integration up, ensure that you have registered Siebel CRM and validated the integration between the applications.

For more information, see:

- *Implement CX Industries Framework* on My Oracle Support, Document ID 2720527.1.
- *Integrate Launch with Siebel CRM* on My Oracle Support, Document ID 2826909.1.

Integrate Launch with Billing and Revenue Management (BRM)

Take advantage of the prebuilt integration between Launch and Billing and Revenue Management, specifically the Pricing Design Center (PDC). Publish product offerings with charges and discounts in PDC and BRM to perform rating and billing.

Set Up the Integration

To set this integration up, ensure that you have registered Buying Experience and validated the integration between the applications.

For more information on integration, see:

- *Implement CX Industries Framework* on My Oracle Support, Document ID 2720527.1.
- *Oracle Communications Pricing Design Center (PDC) 12.0 Patch Set*, on Oracle Help Center.

Integrate Launch with Third-Party AI and ML Insight Providers

Third-party AI/ML application integration allows the alignment of your business KPIs at an offer's design time to bring in predictability in the business outcomes.

Take advantage of the prebuilt integration between Oracle Launch Experience and third-party AI/ML insight providers. Enable a data-driven experience for your product managers and product marketers.

While creating a new offer, as a product manager you can specify expected targets and submit the new offer for analysis. Your targets can include increase in revenue, increase in margin, subscriber growth, and reduction in churn percentage.

The third-party AI/ML application returns:

1. The extent to which the new offer has the potential to satisfy the target.
2. Recommend alternate price points for the offer constituents. They are a better alternative to the original offer in terms of satisfying or beating the mentioned targets.

Third-party AI/ML application integration provides product managers with real-time insight into the projected subscription sales for released offers. It also provides a dashboard containing the offer's predicted revenue generation, channel performance, campaign performance, and subscriber penetration metrics.

Third-party AI/ML applications generate ratings between 1 and 5 stars and displays on the offer dashboard. The ratings represent the revenue from sales of the offer. Ratings are based on historical sales data and prediction of expected sales in the future.

Product managers can use this insight to:

- Act on offers that don't show traction in the marketplace.
- Modify offer constituents or price to improve uptake.
- Perform more marketing programs to further promote the product on various channels.
- Extend the validity of offers that show high potential for sales.
- Easily identify offers that have shown high purchases in the past and create similar offers replicating their success.

Setting Up the Integration

To set this integration up, ensure that you have registered the 3rd-party AI/ML application and validated the integration between the applications.

For more information, see:

- *Implement CX Industries Framework* on My Oracle Support, Document ID 2720527.1.
- *Integrate Data Driven Experience in Launch* on My Oracle Support, Document ID 2858439.1.

10 Appendix

Predefined Role Privilege for Entities

Use this topic to understand the privileges associated with the catalog entities in Launch application. When you create or configure roles, you can implement these functional security privileges to restrict access to the UI depending on the role type. See

Privileges are roles that combine the functional privilege for an individual task or duty with the relevant data security policies. Functions that aggregate privileges might grant access to include task flows, application pages, work areas, dashboards, reports, batch programs, and so on. In this topic, the privileges are for entities in the Launch application.

You can search for details regarding the predefined roles and privileges under **Tools > Security Console > Roles**. See *Guidelines for Configuring Security in Oracle Applications Cloud* for more information on configuring security.

Initiative

An initiative encapsulates product offer and its entities. You can use an initiative or project to publish definitions of all catalog entities to spoke systems.

Privilege Name	Privilege Code	Privilege Description
View Offer Project	ATC_LAUNCHX_VIEW_OFFER_PROJECT	Enables you to view product offer projects.
Create Offer Project	ATC_LAUNCHX_CREATE_OFFER_PROJECT	Enables you to create, clone, and revise product offer projects.
Manage Offer Project	ATC_LAUNCHX_MANAGE_OFFER_PROJECT	Enables you to update product offer projects.
Delete Offer Project	ATC_DELETE_OFFER_PROJECT_PRIV	Enables you to delete product offer projects.
Manage Offer Project Publish	ATC_LAUNCHX_MANAGE_OFFER_PROJECT_PUBLISH	Enables you to publish product offer projects.
View Offer Project Publish Status	ATC_LAUNCHX_VIEW_OFFER_PROJECT_PUBLISH_STATUS	Allows GET on the publish status. Enables you to view the status of the publish project job

Service Specifications

Service specifications reflect the characteristics associated to a service, for example, data service, or a voice service.

Privilege Name	Privilege Code	Privilege Description
View Service Specification	ORA_ATC_VIEW_SERVICE_SPECIFICATION	Enables you to view service specification.

Privilege Name	Privilege Code	Privilege Description
Create Service Specification	ORA_ATC_CREATE_SERVICE_SPECIFICATION	Enables you to create service specification.
Manage Service Specification	ORA_ATC_MANAGE_SERVICE_SPECIFICATION	Enables you to edit the created service specification.
Delete Service Specification	ORA_ATC_DELETE_SERVICE_SPECIFICATION	Enables you to delete the service specification.

Usage Specifications

Usage specifications refer to the usage characteristics of a service, for example, you can set up a price of $50 for 100 GB of data usage, where the rates are based on phone call origin and destination.

Privilege Name	Privilege Code	Privilege Description
View Usage Specification	ORA_ATC_VIEW_USAGE_SPECIFICATION	Enables you to view usage specification.
Create Usage Specification	ORA_ATC_CREATE_USAGE_SPECIFICATION	Enables you to create usage specification.
Manage Usage Specification	ORA_ATC_MANAGE_USAGE_SPECIFICATION	Enables you to edit the created usage specification.
Delete Usage Specification	ORA_ATC_DELETE_USAGE_SPECIFICATION	Enables you to delete the usage specification.

Customer Profile Specifications

Customer profile specifications refer to the customer categories based on different characteristics, such as gold, platinum. You can provide special rates to customers based on these categories.

Privilege Name	Privilege Code	Privilege Description
View Customer Profile Specification	ORA_ATC_VIEW_CUSTOMER_PROFILE_SPECIFICATION	Enables you to view customer profile specification.
Create Customer Profile Specification	ORA_ATC_CREATE_CUSTOMER_PROFILE_SPECIFICATION	Enables you to create customer profile specification.
Manage Customer Profile Specification	ORA_ATC_MANAGE_CUSTOMER_PROFILE_SPECIFICATION	Enables you to edit the customer profile specification.
Delete Customer Profile Specification	ORA_ATC_DELETE_CUSTOMER_PROFILE_SPECIFICATION	Enables you to delete the customer profile specification.

Product Specifications

Product specification consists of detailed description of tangible or intangible objects that are made externally available through the product offer. These are the attributes associated with a product offer and define the technical aspects of the product offer.

Privilege Name	Privilege Code	Privilege Description
View Product Type	ATC_LAUNCHX_VIEW_PRODUCT_TYPE	Enables you to view a single product specification as well as the complete product specification list.
Create Product Type	ATC_LAUNCHX_CREATE_PRODUCT_TYPE	Enables you to create product specifications.
Manage Product Type	ATC_LAUNCHX_MANAGE_PRODUCT_TYPE	Enables you to update product specifications.
Delete Product type	ATC_LAUNCHX_DELETE_PRODUCT_TYPE	Enables you to delete product specifications.

Custom Profile Specifications

Privilege Name	Privilege Code	Privilege Description
View Custom Profile Specification	ORA_ATC_VIEW_CUSTOM_PROFILE_SPECIFICATION	Enables you to view custom profile specification.
Manage Custom Profile Specification	ORA_ATC_MANAGE_CUSTOM_PROFILE_SPECIFICATION	Enables you to create, edit or delete the custom profile specification.

Balance Element

Balance element is the unit of measure for elements whose usage is measured. It can be of type currency or noncurrency.

Privilege Name	Privilege Code	Privilege Description
View Balance Element	ORA_ATC_VIEW_BALANCE_ELEMENT	Enables you to view balance element.
Manage Balance Element	ORA_ATC_MANAGE_BALANCE_ELEMENT	Enables you to create, edit, or delete balance element.

Price List

A price list is a set of standard prices for products and services. You can use multiple price lists to offer different prices for the same product.

Privilege Name	Privilege Code	Privilege Description
Create Price List	ORA_ATC_CREATE_PRICELIST	Enables you to create a price list for a given currency or an account.
View Price List	ORA_ATC_VIEW_PRICELIST	Enables you to view a price list for a given currency or an account.
Manage Price List	ORA_ATC_MANAGE_PRICELIST	Enables you to update or delete a price list for a given currency or an account.

Offer

An offer is a sellable item that's orderable from the provider of the catalog or can also be tracked as an asset. An offer can be a single offer or a bundle offer.

Privilege Name	Privilege Code	Privilege Description
View Offer	ATC_LAUNCHX_VIEW_OFFER	Enables you to view a single product offer as well as the complete product offer list.
Create Offer	ATC_LAUNCHX_CREATE_OFFER	Enables you to create, clone, and revise a product offer.
Manage Offer	ATC_LAUNCHX_MANAGE_OFFER	Enables you to update a product offer.
Delete Offer	ATC_DELETE_OFFER_PRIV	Enables you to delete a product offer.
Revise Offer	ATC_LAUNCHX_REVISE_OFFER	Enables you to create revisions of a product offer.
Retire Offer	ATC_LAUNCHX_RETIRE_OFFER	Enables you to retire a product offer.
Obsolete Offer	ATC_LAUNCHX_OBSOLETE_OFFER	Enables you to obsolete a product offer.
View System Offer	ORA_ATC_VIEW_SYSTEM_OFFER_PRIV	Gives you access to view system offers.
Manage System Offer	ORA_ATC_MANAGE_SYSTEM_OFFER_PRIV	Lets you update system offers.
Delete System Offer	ORA_ATC_DELETE_SYSTEM_OFFER_PRIV	Lets you delete system offers.

Product Line

Product lines group similar product offers.

Privilege Name	Privilege Code	Privilege Description
View Product Line	ATC_LAUNCHX_VIEW_PRODUCT_LINE	Allows GET operation on product lines. Enables you to view product lines.
Manage Product Line	ATC_LAUNCHX_MANAGE_PRODUCT_LINE	Enables you to create, update, and delete product lines.

Catalog

A catalog is a collection of product offers intended for a specific distribution channel. A catalog can be associated with a region, a channel, or purpose.

Privilege Name	Privilege Code	Privilege Description
View Offer Catalog	ATC_LAUNCHX_VIEW_OFFER_CATALOG	Enables you to view the list of product catalog and categories.
Manage Offer Catalog	ATC_LAUNCHX_MANAGE_OFFER_CATALOG	Enables you to create, update, and retire catalog or categories.
Delete Offer Catalog	ATC_DELETE_OFFER_CATALOG_PRIV	Enables you to delete a product catalog or category.

Life Cycle Configuration

Life cycle configuration enables you to define and work with custom life cycle statuses along with the lifecycle statuses that are shipped with Launch Experience.

Privilege Name	Privilege Code	Privilege Description
View Life Cycle Configuration	ORA_ATC_VIEW_LIFECYCLE_CONFIGURATION	Enables you to view lifecycle configuration.
Manage Life Cycle Configuration	ORA_ATC_MANAGE_LIFECYCLE_CONFIGURATION	Enables you to create, edit or delete the lifecycle configuration.

Pricing Logic Algorithm Type

The Pricing Logic Algorithm specifications use black-box pricing algorithm parameter values to calculate the final price of the product.

Privilege Name	Privilege Code	Privilege Description
View Pricing Logic Algorithm Type	ATC_VIEW_PRICING_LOGIC_ALGORITHM_TYPE_PRIV	Enables you to view pricing logic algorithm types.

Privilege Name	Privilege Code	Privilege Description
Manage Pricing Logic Algorithm Type	ATC_MANAGE_PRICING_LOGIC_ALGORITHM_TYPE_PRIV	Enables you to update pricing logic algorithm types.
Delete Pricing Logic Algorithm Type	ATC_DELETE_PRICING_LOGIC_ALGORITHM_TYPE_PRIV	Enables you to delete pricing logic algorithm types.

Product Offer Price

Product offer pricing consists of adding different pricing schemes to an offer during its creation, such as simple pricing and advanced pricing.

Privilege Name	Privilege Code	Privilege Description
View Product Offer Price	ATC_VIEW_PRODUCT_OFFER_PRICE_PRIV	Enables you to view product offer price.
Create Product Offer Price	ATC_CREATE_PRODUCT_OFFER_PRICE_PRIV	Enables you to view, create, and manage product offer price.
Manage Product Offer Price	ATC_MANAGE_PRODUCT_OFFER_PRICE_PRIV	Enables you to update product offer price.
Delete Product Offer Price	ATC_DELETE_PRODUCT_OFFER_PRICE_PRIV	Enables you to delete product offer price.

Tax Service Provider

A tax service provider represents simple and seller specific tax codes that are used in product offer pricing.

Privilege Name	Privilege Code	Privilege Description
View Tax Service Provider	ORA_ATC_VIEW_TAX_SERVICE_PROVIDER	Enables you to view tax service provider.
Create Tax Service Provider	ORA_ATC_CREATE_TAX_SERVICE_PROVIDER	Enables you to create tax service provider.
Manage Tax Service Provider	ORA_ATC_MANAGE_TAX_SERVICE_PROVIDER	Enables you to edit the tax service provider.
Delete Tax Service Provider	ORA_ATC_DELETE_TAX_SERVICE_PROVIDER	Enables you to delete the tax service provider.

Promotion

Promotion enables you to provide additional discounts, vouchers, bonuses or gifts to the customer who meets the pre-defined criteria.

Privilege Name	Privilege Code	Privilege Description
Create Promotion	ORA_ATC_CREATE_PROMOTION	Enables you to create TMF 671 promotion object.
View Promotion	ORA_ATC_VIEW_PROMOTION	Enables you to view TMF 671 promotion object.
Update Promotion	ORA_ATC_UPDATE_PROMOTION	Enables you to update TMF 671 promotion object.
Delete Promotion	ORA_ATC_DELETE_PROMOTION	Enables you to delete TMF 671 promotion object.

Import

Import enables you to bring third-party catalog entities into Launch Experience, either through APIs or through UI. Entities include catalogs, categories, product lines, product specifications, product offers, bundle offers, product offer prices and their alterations, rules, terms, and attribute translations.

Privilege Name	Privilege Code	Privilege Description
View Import Status	ATC_VIEW_IMPORT_STATUS_PRIV	Allows GET operation on the import job. Enables you to view the import job status.
Manage Import	ATC_MANAGE_IMPORT_PRIV	Enables you to import catalog entities by creating an import offer job.

Export

The export functionality enables you to export specific resource types, such as, catalogs, categories, initiative, product lines, product specifications, product offers, bundle offers, product offer prices, and so on from Launch Experience through REST APIs and retrieve the exported data files from a known location.

Privilege Name	Privilege Code	Privilege Description
Create Export Job	ORA_ATC_CREATE_EXPORT_JOB	Enables you to create an export job to export catalog entities.
View Export Job	ORA_ATC_VIEW_EXPORT_JOB	Enables you to view an export job to export catalog entities.
Manage Export Job	ORA_ATC_MANAGE_EXPORT_JOB	Enables you to delete or update an export job to export catalog entities.

Digital Content

You can manage your content such as images, collateral and other documents, agreements for an offer, or add images to the SKU for a product specification.

Privilege Name	Privilege Code	Privilege Description
View Offer Attachment	ATC_VIEW_ATTACHMENT_PRIV	Allows GET operation on the attachments. Allows download of any file attachment from the content repository
Manage Offer Attachment	ATC_LAUNCHX_MANAGE_OFFER_ATTACHMENT	Allows POST or PUT operation on the attachments. Allows upload of any file attachment from the content repository

Workspace

Privilege Name	Privilege Code	Privilege Description
View Workspace	ATC_LAUNCHX_VIEW_WORKSPACE	Allow GET operation on destination. Enables you to create a destination.
Manage Workspace	ATC_LAUNCHX_MANAGE_WORKSPACE	Enables you to create, update, and delete destinations.

Initiative Approval Action

Initiative Approval Action enables you to Submit an Initiative for Approval flow.

Privilege Name	Privilege Code	Privilege Description
Submit Initiative Approval Action	ORA_ATC_SUBMIT_INITIATIVE_APPROVAL_ACTION	Enables you to submit initiative approval action.
Submit Initiative Approver Action	ORA_ATC_SUBMIT_INITIATIVE_APPROVER_ACTION	Enables you to submit initiative approver action.

Rules

Rules are conditions that you can set for product offers, product lines, and product specifications. You can set rules for eligibility or compatibility of offers or specifications, downgrading or upgrading terms, or for recommendations.

Privilege Name	Privilege Code	Privilege Description
View Rules	ORA_ATC_VIEW_RULES_PRIV	Allows access to rules list and single rules view.
Manage Rules	ORA_ATC_MANAGE_RULES_PRIV	Allows updating rules.

Privilege Name	Privilege Code	Privilege Description
Delete Rules	ORA_ATC_DELETE_RULES_PRIV	Allows deletion of rules

User Interface Task Cards

Task cards provide information or an entry point into task details

Privilege Name	Privilege Code	Privilege Description
View Offer Management Pane	ORA_ATC_VIEW_RULES_PRIV	Allows viewing Offer Management card.
View Promotion Management Pane	ORA_ATC_VIEW_PROMOTION_MANAGEMENT_PANE_PRIV	Allows viewing Promotion Management card.
View Digital Content Tile	ORA_ATC_VIEW_DIGITAL_CONTENT_TILE_PRIV	Allows viewing of Digital Content card.
View Catalogs and Categories Tile	ORA_ATC_VIEW_CATALOGS_CATEGORIES_TILE_PRIV	Allows viewing Catalogs and Categories card.
View Initiatives Tile	ORA_ATC_VIEW_INITIATIVES_TILE_PRIV	Allows viewing Initiatives card.
View Lifecycle Status Tile	ORA_ATC_VIEW_LIFECYCLE_STATUS_TILE_PRIV	Allows viewing Lifecycle Status tile.
View Price Lists Tile	ORA_ATC_VIEW_PRICE_LISTS_TILE_PRIV	Allows viewing Price Lists tile.
View Product Lines Tile	ORA_ATC_VIEW_PRODUCT_LINES_TILE_PRIV	Allows viewing product lines tile.
View Job Management Tile	ORA_ATC_VIEW_JOB_MANAGEMENT_TILE_PRIV	Allows viewing Job Management tile.
View Rules Tile	ORA_ATC_VIEW_RULES_TILE_PRIV	Allows viewing rules tile.
View Custom Profile Tile	ORA_ATC_VIEW_CUSTOM_PROF_TILE_PRIV	Allows viewing custom profile tile.
Manage Confirm action	ORA_ATC_MANAGE_CONFIRM_ACTION_PRIV	Allows confirming the design completion of an entity.
View Finance Plan Tile	ORA_ATC_VIEW_FINANCE_PLAN_TILE_PRIV	Enables you to view the finance plan configuration tile.
View System Offer Tile	ORA_ATC_VIEW_SYSTEM_OFFER_TILE_PRIV	Enables you to view the system offer configuration tile.

Return Checklist

Return checklist templates to support the product return scenario.

Privilege Name	Privilege Code	Privilege Description
View Return Checklist Tile	ORA_ATC_VIEW_RET_CHECK_TILE_PRIV	Enables viewing the return checklist tile in the user interface.
View Return Checklist	ORA_ATC_VIEW_RET_CHECK_LIST_PRIV	Enables access to viewing return checklist.
Delete Return Checklist	ORA_ATC_DELETE_RET_CHECK_LIST_PRIV	Enables deleting the return checklist.
Manage Return Checklist	ORA_ATC_MANAGE_RET_CHECK_LIST_PRIV	Enables updating the return checklist.

Entitlement

Entitlements are part of the sales / Service agreement which includes information about the special pricing which can be applied against the sales products.

Privilege Name	Privilege Code	Privilege Description
View Entitlement Templates	ORA_ATC_VIEW_ENTITLEMENT_TEMP_PRIV	Allows access to view Entitlement templates
Manage Entitlement Templates	ORA_ATC_MANAGE_ENTITLEMENT_TEMP_PRIV	Allows update of Entitlement templates
Delete Entitlement Templates	ORA_ATC_DELETE_ENTITLEMENT_TEMP_PRIV	Allows deletion of Entitlement Templates
View Entitlement Templates Tile	ORA_ATC_VIEW_ENTITLEMENT_TEMP_TILE_PRIV	Allows viewing of Entitlement Templates tile in User Interface

Others

Privilege Name	Privilege Code	Privilege Description
Change Offer Lifecycle status	ATC_CHANGE_OFFER_LIFECYCLE_STATUS_PRIV	Enables you to update an offer lifecycle status.
View Functional Setup Area	ATC_LAUNCHX_VIEW_FUNCTIONAL_SETUP_AREA	Allows access to the Functional Set up area.
View Product Catalog Items	ATC_VIEW_PRODUCT_CATALOG_ITEMS_PRIV	Enables you to view product catalogs. Also enables you to get the Launch icon displayed on the Oracle Fusion springboard.
Manage Catalog Extension	ORA_ATC_MANAGE_CATALOG_EXTENSION_PRIV	Enables you to create and update catalog extensions.
View Catalog Extension	ORA_ATC_VIEW_CATALOG_EXTENSION_PRIV	Enables you to view catalog extensions.

Privilege Name	Privilege Code	Privilege Description
View Extension Tile	ORA_ATC_EXTENSION_TILE	Enables you to view extension tiles in the user interface.

Business Configuration

Customers can extend the default custom profile spec in order create their own business configuration by adding additional characteristics or additional characteristics value.

Privilege Name	Privilege Code	Privilege Description
View Business Configuration	ORA_ATC_VIEW_BUSINESS_CONFIG_PRIV	Allows access to view the business configuration.
Manage Business Configuration	ORA_ATC_MANAGE_BUSINESS_CONFIG_PRIV	Enables you to update the business configuration.
Delete Business Configuration	ORA_ATC_DELETE_BUSINESS_CONFIG_PRIV	Enables you to delete a business configuration.
View Business Configuration Tile	ORA_ATC_VIEW_BUS_CONF_TILE_PRIV	Allows viewing of Business Configuration Tile.

View Offer Comparison

Customers can compare offers.

Privilege Name	Privilege Code	Privilege Description
View Offer Comparison	ORA_ATC_VIEW_OFFER_COMPARISON_PRIV	Enables you to view and compare differences between offers.

Entity Profile

Customers can view and manage entity profiles.

Privilege Name	Privilege Code	Privilege Description
View Entity Profile	ORA_ATC_VIEW_ENTITY_PROFILE_PRIV	Enables you to view entity profiles.
Manage Entity Profile	ORA_ATC_MANAGE_ENTITY_PROFILE_PRIV	Enables you to update entity profiles.
Delete Entity Profile	ORA_ATC_DELETE_ENTITY_PROFILE_PRIV	Enables you to delete entity profiles.
View Entity Profile Configuration Tile	ORA_ATC_VIEW_ENTITY_PROFILE_TILE_PRIV	Enables you to view the entity profile configuration tile.

Publish Job

Customers can view published jobs.

Privilege Name	Privilege Code	Privilege Description
View Publish Job Tile	ORA_ATC_VIEW PUB_JOB_TILE_PRIV	Enables you to view publish job tile.

Finance Plan

Finance plans allow you to factor in interest based installment plans with amortization details in addition to the current 0% interest based installment plans.

Privilege Name	Privilege Code	Privilege Description
View Finance Plan	ORA_ATC_VIEW_FINANCE_PLAN_PRIV	Gives you access to view finance plans.
Manage Finance Plan	ORA_ATC_MANAGE_FINANCE_PLAN_PRIV	Enables you to update finance plans.
Delete Finance Plan	ORA_ATC_DELETE_FINANCE_PLAN_PRIV	Enables you to delete finance plans.

Zone Value Map

Customers can create and manage zones and value maps based on zones which can be used in zone based pricing.

Privilege Name	Privilege Code	Privilege Description
View Zone Value Map	ORA_ATC_VIEW_ZONE_VALUEMAP_PRIV	Gives you access to view the zone value map.
Manage Zone Value Map	ORA_ATC_MANAGE_ZONE_VALUEMAP_PRIV	Enables you to update the zone value map.
Delete Zone Value Map	ORA_ATC_DELETE_ZONE_VALUEMAP_PRIV	Enables you to delete the zone value map.
View Bulk Update Job Tile	ORA_ATC_VIEW_BULK_UPDATE_JOB_TILE_PRIV	Enables you to view the bulk update job tile.

Bulk Update Job

Bulk update enables customers to select multiple records of an entity and update its attributes.

Privilege Name	Privilege Code	Privilege Description
View Bulk Update Job	ORA_ATC_VIEW_BULK_UPDATE_JOB_PRIV	Gives you the privilege to view a bulk update job.

Privilege Name	Privilege Code	Privilege Description
Manage Bulk Update Job	ORA_ATC_MANAGE_BULK_UPDATE_JOB_PRIV	Gives you the privilege to manage a bulk update job.
Manage Offer Bulk Update	ORA_ATC_MANAGE_OFFER_BULK_UPDATE_PRIV	Gives you the privilege to trigger an offer bulk update.

User Interface Task Cards

Task cards provide information or an entry point into task details.

Privilege Name	Privilege Code	Privilege Description
View Bulk Update Job Tile	ORA_ATC_VIEW_BULK_UPDATE_JOB_TILE_PRIV	Enables you to view the bulk update job tile.

Predefined Job and Duty Roles

Use this topic to know about the predefined job roles and duty roles available in the Launch application.

Job roles represent the jobs that the users perform in an organization and duty roles determine the tasks that the users can perform as part of their jobs. You can search for these roles under **Tools > Security Console > Roles**.

This table provides a list of the seeded job roles and the associated duty roles in the Launch application. You can also create job roles as necessary using Security Console. See *Guidelines for Configuring Security in Oracle Applications Cloud* for more information about configuring security.

Job Role	Job Role Code	Description	Job Function	Duty Role
Communications Catalog Administrator	ORA_ATC_COMMUNICATIONS_CATALOG_ADMINISTRATOR_JOB	A super user who performs application configuration set up, data loads, functional setups, and all offer lifecycle management tasks.	• Create, update, and delete of all catalog definitions • Override the lifecycle status for catalog definitions • Perform all application administrator set ups • Perform importing of catalog definitions • Create and manage initiatives for publish	• Offer Inquiry • Offer Management • Offer Catalog Entity Deletion • Workbench Inquiry • Workbench Pane Management • Offer Promotion Pane Management • Administration Pane Management • Workbench Management • Offer Functional Setups Inquiry

Job Role	Job Role Code	Description	Job Function	Duty Role
				• Offer Functional Setups Management • Offer Catalog Item Management • Offer Catalog Item Inquiry • Change Offer Lifecycle Status privilege • Catalog Import Export Administration • Lifecycle Status Management • Manage Entity Confirm Action
Communications Catalog Product Manager	ORA_ATC_ COMMUNICATIONS_ CATALOG_PRODUCT_ MANAGER_JOB	A primary user who ensures that a product offer gets created, tested, and shipped on schedule and meets specifications. A member of either the marketing organization or the product management team that launches net new product offers and more of the same offers. Also performs functional setup tasks and creates product specifications.	• Create and update catalog definitions but can't delete them • Perform all application administrator setups • Perform importing of catalog definitions • Create and manage initiatives for publish	• Offer Inquiry • Offer Management • Offer Catalog Item Inquiry • Offer Catalog Item Management • Workbench Inquiry • Offer Promotion Pane Management • Offer Functional Setups Inquiry • Manage Entity Confirm Action
Communications Marketing Manager	ORA_ATC_ COMMUNICATIONS_ MARKETING_MANAGER_ JOB	Ensures that a product offer gets created, tested, and shipped on schedule and meets specifications. A member of marketing organization that launches more of the same offers.	• Create and update offers but can't delete them • Can't override the lifecycle status of an offer • Can't create or manage product types • Can't perform application administrator setups of catalogs, categories, initiatives, and product lines, but not deletes • Can't import third-party catalogs • Can't create and manage initiatives for publish	• Offer Inquiry • Offer Management • Offer Catalog Item Inquiry • Offer Catalog Item Management • Workbench Inquiry • Offer Promotion Pane Management • Offer Functional Setups Inquiry • Manage Entity Confirm Action

Job Role	Job Role Code	Description	Job Function	Duty Role
Communications Catalog Viewer	ORA_ATC_ COMMUNICATIONS_ CATALOG_VIEWER_JOB	An oversight or management user that accesses the application in a restricted and read-only manner.	• Can't create, update, and delete any entity or perform administrator setup • View all offers and product types • View application administrator set ups of catalogs, categories, initiatives, product lines • Can't import third-party catalogs • View initiatives	• Offer Inquiry • Offer Catalog Item Inquiry • Workbench Inquiry • Offer Functional Setups Inquiry • Workbench Pane Management • Offer Promotion Pane Management
Communications Technical Specialist	ORA_ATC_ COMMUNICATIONS_ TECHNICAL_SPECIALIST_ JOB	Individual responsible for managing a technical portfolio, including creating and managing specifications.	Create and manage product specifications Can't delete product specifications	• Workbench Inquiry • Workbench Management • Offer Inquiry • Offer Functional Setups Inquiry • Catalog Content Management • Workbench Pane Management • Manage Entity Confirm Action

These job roles also have duty roles associated to them. You use these duty roles to grant access to tasks that users can perform as part of their jobs.

This table provides information about the available duty roles in Launch application:

Duty Roles	Duty Role Code	Description	Privileges
Offer Inquiry	ORA_ATC_OFFER_INQUIRY_DUTY	Queries and views product offers in the enterprise	• View Offers Work Area • View Offer • View Product Offer Price • View Offer Attachment • View Product Catalog Items • View Promotion • Submit Initiative Approval Action • View Rules • View Bulk Update Job

Duty Roles	Duty Role Code	Description	Privileges
Offer Management	ORA_ATC_OFFER_MANAGEMENT_DUTY	Creates and manages catalogs, categories, product offers including attributes, components, rules, terms, pricing and catalog assignments, and lifecycle statuses	• Create Offer • Manage Offer • Publish Offer • Revise Offer • Retire Offer • Obsolete Offer • Create Product Offer Price • Manage Product Offer Price • Manage Offer Attachment • View Product Catalog Items • Create Promotion • View Promotion • Manage Promotion • Submit Initiative Approval Action • Manage Rules • Manage Offer Bulk Update • Manage Bulk Update Job
Offer Catalog Entity Deletion	ORA_ATC_OFFER_CATALOG_ENTITY_DELETION_DUTY	Deletes product offers, catalogs, categories, product types after they have become obsolete and are no longer needed	• Delete Offer • Delete Offer Catalog • Delete Product Type • Delete Product offer Price • Delete Offer Project • Delete Pricing Logic Algorithm Type • View Product Catalog Items • Delete Rules • Delete Return Checklist • Delete Entitlement Templates • Delete Entity Profile • Delete Finance Plan • Delete System Offer • Delete Zone Value Map
Workbench Inquiry	ORA_ATC_WORKBENCH_INQUIRY_DUTY	Views product types and pricing logic algorithm types that are used to define product offers and product offer prices	• View Product Type • View Pricing Logic Algorithm Types • View Product Catalog Items • View Service Specification • View Usage Specification

Duty Roles	Duty Role Code	Description	Privileges
			- View Customer Profile Specification - View Custom Profile Specification
Workbench Management	ORA_ATC_WORKBENCH_ MANAGEMENT_DUTY	Manages product types and pricing logic algorithm types that are used to define product offers and product offer prices	- Create Product type - Manage Product Type - Manage Pricing Logic Algorithm Type - View Product Catalog Items - View Service Specification - Create Service Specification - Manage Service Specification - Delete Service Specification - View Usage Specification - Create Usage Specification - Manage Usage Specification - Delete Usage Specification - View Customer Profile Specification - Create Customer Profile Specification - Manage Customer Profile Specification - Delete Customer Profile Specification - View Custom Profile Specification - Create Custom Profile Specification - Manage Custom Profile Specification - Delete Custom Profile Specification
Offer Functional Setups Inquiry	ORA_ATC_OFFER_FUNCTIONAL_ SETUPS_INQUIRY_DUTY	Views functional setups including workspaces, catalogs, categories, offer settings, product lines, and so on.	- View Functional Setup Area - View Workspace - View Offer Catalog - View Product Line - View Import Status - View Export Status - View Offer Project - View Offer Project Publish Status

Duty Roles	Duty Role Code	Description	Privileges
			• View Product Catalog Items • View Multiple Price List • View Tax Service Provider • View Balance Element • View Life Cycle Configuration • View Rules • View Return Checklist • View Entitlement Templates • Manage Entity Profile • View Finance Plan • View System Offer • View Zone Value Map
Offer Functional Setups Management	ORA_ATC_OFFER_FUNCTIONAL_SETUPS_MANAGEMENT_DUTY	Performs functional setups including managing workspaces, catalogs, categories, offer settings, and product lines.	• Manage Workspace • Manage Offer Catalog • Manage Product Line • Manage Import • Manage Export • Create Offer project • Manage Offer Project Publish • View Product Catalog Items • Create Multiple Price List • View Multiple Price List • Manage Multiple Price List • Create Tax Service Provider • Manage Tax Service Provider • Delete Tax Service Provider • View Balance Element • Manage Balance Element • View Life Cycle Configuration • Manage Life Cycle Configuration • Manage Rules • Manage Return Checklist • Manage Entitlement Templates • Manage Finance Plan • Manage System Offer • Manage Zone Value Map

Duty Roles	Duty Role Code	Description	Privileges
Offer Catalog Item Inquiry	ORA_ATC_OFFER_CATALOG_ITEM_INQUIRY	Views items and item-related information for offer catalog	• Manage Item Catalog • Manage Item Global Search • View Item • View Item Organization Association • View Item Relationship • Browse Item • Create Item Structure • Monitor Item Summary • Monitor Item Work area • Monitor Item Dashboard
Offer Catalog Item Management	ORA_ATC_OFFER_CATALOG_ITEM_MANAGEMENT	Creates and manages items and item-related information for offer catalog	• Browse Item • Configure Item Extended Attributes • Create Item Structure • Manage Item • Manage Item Catalog • Manage Item Lifecycle Phase Changes • Manage Item Organization Association • Manage Item Relationships • Manage Item Status Change • Monitor Item Summary • Monitor Item Work area • Monitor Item Dashboard • View Item • View Item Organization Association • View Item Relationship
Workbench Pane Management	ORA_ATC_WORKBENCH_PANE_MANAGEMENT	A user who manager the workbench contents in the user interface.	• View Workbench Pane • View Digital Content Tile
Offer Promotion Pane Management	ORA_ATC_OFFER_PROMOTION_PANE_MANAGEMENT	A user who manages the Offer and Promotion pane contents in the user interface.	• View Offer Management Pane • View Promotion Management Pane • View Digital Content Tile • View Bulk Update Job Tile

Duty Roles	Duty Role Code	Description	Privileges
Administration Pane Management	ORA_ATC_ADMINISTRATION_PANE_MANAGEMENT	A user who manages the Administration contents in the user interface.	• View Digital Content Tile • View Catalogs and Categories Tile • View Initiatives Tile • View Lifecycle Status Tile • View Price Lists Tile • View Product Lines Tile • View Job Management Tile • View Rules Tile • View Return Checklist Tile • View Entitlement Templates • View Publish Job Tile • View Custom Profile Tile • View Business Configuration Tile • View Entity Profile Configuration Tile • View Finance Plan Tile • View System Offer Tile
Catalog Content Management	ORA_ATC_CATALOG_CONTENT_MANAGEMENT	A user who manages all types of contents and collateral.	• View Offer Attachment • Manage Offer Attachment
Catalog Extension Management	ORA_ATC_CATALOG_EXTENSION_MANAGEMENT_DUTY	A user who manages the catalog schema extension.	• Manage Extensible Object • Administer Sandbox • Manage Catalog Extension • View Catalog Extension • View Extension Tile
Lifecycle Status Management	ORA_ATC_OFFER_LIFECYCLE_STATUS_MANAGEMENT	A user who manages the entity life cycle configuration.	Manage Lifecycle Status
Manage Entity Confirm Action	ORA_ATC_CONFIRM_ACTION	A user who confirms the design completion of an entity.	Manage Confirm Action.

Design-time Restriction Properties to Create Entity Profiles

Use this topic to view the design-time restriction properties you can use to create entity profiles.

Here's a list of design-time restriction properties:

- Rule: Defines the type of restriction to be used as a rule while creating an entity profile
- Condition : Defines specific conditions where restrictions are allowed. For example, apply restriction for product offers of type device.
- Restriction : Defines a simple restriction where only one feature and one attribute are mapped
- Expression : Indicates the mapping attribute name used to restrict the entity
- customProfileSpecChar name of type FEATURE : Indicates the name used for FEATURE type to define an entity profile
- customProfileSpecChar name of type ATTRIBUTE : Indicates the name used for ATTRIBUTE type to define an entity profile
- FEATURE : Collection of rules, which is the starting point of rule validation
- ATTRIBUTE : Construct by which a certain attribute name is associated with some values
- EXCLUDES : Excludes the mentioned feature or attribute property from the payload and validates others
- ALLOWED : Allows the mentioned feature or attribute property from the payload and validates

This table lists the different values you can use to create restrictions.

Rule	Condition	Restriction	Expression	customProfileSpecCha name of type FEATURE	customProfileSpecChar name of type ATTRIBUTE
Restrict only 1 commitment term for Product Offering entity of package type	Product type is "Package"	Restrict only 1 commitment term	productOfferingTerm[@	info_productType_bundle	requires_term_type_commitmentTerm
Restrict Payment terms	-	Restrict only payment term	productOfferingTerm[@	rule 1_term_type_paymentTerm	exclude_term_type_paymentTerm
Restrict Charging terms	-	-	customProfileSpec.[id,, 'CustomProfileSpecifica	rule 1_term_type	exclude_term_type
Restrict Override of commitment term	-	-	-	NA	-
Restrict Override of option group	-	-	choiceRelationship	rule 1_bundledProductOfferin overrideOptionGroup	-
Restrict Storefront Banners	-	Restrict banner	banner	rule 1_productOffering_banner	exclude_productOffering_banner
Restrict Marketing features	-	Restrict marketing feature	marketingFeature	rule 1_productOffering_marketingFeature	exclude_productOffering_marketingFeature
Restrict One time price	-	-	productOfferingPrice.[id,, 'ProductOfferingPriceO	rule1_price_pop_priceType	excludes_price_pop_priceType
Restrict Attribute based pricing	-	restrict attribute based pricing in offer	productOfferingPrice.[id,, 'ProductOfferingPriceO	rule1_price_pop_advancedPriceType_attribute	excludes_price_pop_advancedPriceType_attribute

Rule	Condition	Restriction	Expression	customProfileSpecCha name of type FEATURE	customProfileSpecChar name of type ATTRIBUTE
Restrict Volume / Tiered	-	Restrict volume & tiered discount	productOfferingPrice. [id,, 'ProductOfferingPriceO	rule 1_price_pop_ advancedPriceType_ tiered	excludes_price_pop_ advancedPriceType_ tiered
Restrict Allowances and Overages	-	Restrict Allowance & Overage price	productOfferingPrice. [id,, 'ProductOfferingPriceO	Restrict Allowance_ price_pop_ advancedPriceType_ allowance	excludes_price_pop_ advancedPriceType_ allowance
Restrict Time limited discount	-	Restrict time limited discount	productOfferingPrice. [id,, 'ProductOfferingPriceO	rule 1_price_pop_ alteration_type_ timeLimitedDiscount	requires_price_pop_ alteration_type_ timeLimitedDiscount
Restrict Installment plans	-	Restrict installment plans	productOfferingTerm[c	rule 1_price_pop_ installment	-
Restrict Lease plans	-	Restrict lease plans	productOfferingTerm[c	rule 1_price_pop_lease	excludes_price_pop_ lease
Restrict Primary / Auxiliary service spec in productSpecification	-	Restrict primary spec	serviceSpecification.rol	Product specification rule_info_ serviceSpecification_ role	excludes_info_ serviceSpecification_ role
Restrict Usage spec in productSpecification	-	Restrict usage spec	usageSpecification	Product specification rule_info_ usageSpecification	-
Restrict Customer profile spec in productSpecification	-	Restrict customerProfileSpecific	customerProfileSpecific	Product specification rule_info_ customerProfileSpecific	-
Restrict Aggregate attributes in productSpecification	-	Restrict aggregate attribute	productSpecCharacteris	Product specification rule_ prodSpecAttribute_ characteristics_type	excludes_ prodSpecAttribute_ characteristics_type
Restrict Data types(DECIMAL/ BOOLEAN) in productSpecification	-	Restrict decimal type value	productSpecCharacteris	Product specification rule_ prodSpecAttribute_ characteristics_ valueType	excludes_ prodSpecAttribute_ characteristics_ valueType
Restrict Data types (STRING/NUMBER) in ServiceSpecification	-	Restrict service spec value type	serviceSpecCharacteris	Service specification rule_ serviceSpecAttribute_ characteristics_ valueType	excludes_ serviceSpecAttribute_ characteristics_ valueType
Restrict Device Trade in	-	Restrict device trade in	promotionType	Device tradein rule_promotion_ promotionType_ deviceTradeIn	excludes_promotion_ promotionType_ deviceTradeIn
Restrict reusable price plans	-	Restrict reusable price plans	isReusable	rule 1_price_pop_ reusable	excludes_price_pop_ reusable

Rule	Condition	Restriction	Expression	customProfileSpecCha name of type FEATURE	customProfileSpecChar name of type ATTRIBUTE
Restrict Price Model Adjustments	-	Restrict Fixed discounts/Percentage discounts/Fixed Markup/Percentage Markup/Volume discount/Tiered discount	productOfferingPrice.ad	rule 1_price_pop_alteration_type	excludes_price_pop_alteration_type
Restrict Adjustments per priceList	-	Restrict Adjustments per priceList	pricelist.alteration	rule 1_price_priceList_pop_alteration	excludes_price_priceList_pop_alteration
Restrict Instalment Plan	-	Restrict Instalment Plan	productOfferingTerm.in	rule 1_price_pop_installment_type	excludes_price_pop_installment_type
Restrict reusable adjustment	-	Restrict reusable adjustment	isReusableAdjustment	rule 1_price_pop_alteration_reusable	excludes_price_pop_alteration_reusable
Restrict promotion type	-	Restrict promotion type	promotionType	rule 1_promotion_promotionType	excludes_promotion_promotionType
Restrict Multiple criteria groups	-	Restrict Multiple criteria groups	pattern.criterialLogicalR	rule 1_promotion_event_criteriaGroupLogicalRel	excludes_promotion_event_criteriaGroupLogicalRelationship
Restrict Benefit price type	-	Restrict Benefit price type	pattern.action.actionVa	rule 1_promotion_event_benefit_feeType	excludes_promotion_event_benefit_feeType
Restrict Promotion benefits	-	Restrict Promotion benefits	pattern.action.actionTy	rule 1_promotion_event_benefit_actionType	excludes_promotion_event_benefit_actionType
Restrict Cross Product Discount	-	Restrict Cross Product Discount	promotionType	rule 1_promotion_promotionType_crossProductDiscount	excludes_promotion_promotionType_crossProductDiscount
Restrict Multiple criteria groups	-	Restrict Multiple criteria groups	pattern.criterialLogicalR	rule 1_crossProductDiscount_event_criteriaGroupLogicalRel	excludes_crossProductDiscount_event_criteriaGroupLogicalRelationship
Restrict Benefit price type	-	Restrict Benefit price type	pattern.action.actionVa	rule 1_crossProductDiscount_event_benefit_feeType	excludes_crossProductDiscount_event_benefit_feeType
Restrict Recommendation rules	ProductRule type is Recommendation	Restrict Recommendation rules Effectivity	ruleCondition	Rule1_productRule_ruleType	condition_ruleType_recommendation, requires_productRule_recommendationRule_effectivity
Restrict Compatibility rules	ProductRule type is Compatibility	Restrict Compatibility rules Effectivity	ruleCondition	Rule2_productRule_ruleType	condition_ruleType_compatibility, requires_productRule_compatibilityRule_effectivity

Rule	Condition	Restriction	Expression	customProfileSpecCha name of type FEATURE	customProfileSpecChar name of type ATTRIBUTE
Restrict Migration rules	ProductRule type is Migration	Restrict Migration rules Effectivity	ruleCondition	Rule3_productRule_ ruleType	condition_ruleType_ migration, requires_productRule_ migrationRule_ effectivity
Restrict Eligibility rules	ProductRule type is Eligibility	Restrict eligibility rules Effectivity	ruleCondition	Rule4_productRule_ ruleType	condition_ruleType_ eligibility, requires_productRule_ eligibilityRule_ effectivity
Restrict Eligibility rules	ProductRule type is Eligibility	Restrict eligibility rules for Rule type	ruleCondition.requestC	Rule5_productRule_ ruleType	condition_ruleType_ eligibility, excludes_ productRule_ eligibilityRule_ ruleType
Restrict extended offer type in price rollups	Product type is Discount	Restrict extended offer type in price rollups	restrictBundlePriceRollu	rule1_info_ extendProductType	condition_ extendProductType_ discount excludes_ bundleProductOffering_ restrictBundlePriceRollup
Restrict extended offer type in bundle configuration	Product type is Discount	Restrict extended offer type in bundle configuration	restrictBundleConfig	rule1_info_ extendProductType	condition_ extendProductType_ discount excludes_ bundleProductOffering_ restrictBundleConfig
Restrict extended offer type in Promotion benefits configuration	Product type is Discount	Restrict extended offer type in Promotion benefits configuration	restrictPromotionEvent	rule1_info_ extendProductType	condition_ extendProductType_ discount excludes_promotion_ events
Restrict extended offer type in product rules configuration	Product type is Discount	Restrict extended offer type in product rules configuration	restrictProductRuleEnti	rule1_info_ extendProductType	condition_ extendProductType_ discount excludes_ productRule_entity