## Contents

**Preface**

1 Configure Innovation Management

- An Introduction to Innovation Management
- Checklist to Configure Oracle Innovation Management Cloud
- Deploy Innovation Management

2 Set Up Innovation Management Functional Areas

- Innovation Management Functional Areas
- Configure External Systems for Integration
- Pages and Page Layouts in Application Composer
- Class Management in Oracle Innovation Management
- Innovation Management Lookups
- Configure Planning Periods for Your Portfolio
- Portfolio Metrics
- Set Up Concept Rollups
- Portfolio and Product Rule Sets
- Enable Single Sign-On for Innovation Management
- FAQs on Social Networking in Oracle Innovation Management
3 Configure the User Interface with Application Composer 25
   - Overview of the Application Composer for Oracle PLM Cloud
   - How to Upgrade to an Oracle PLM Cloud Release that Features Application Composer
   - Capabilities of Application Composer per Object Type
   - Getting Started in Application Composer
   - Example of How You Work with Sandboxes in Application Composer
   - Layout Nodes in Application Composer
   - Pages and Page Layouts in Application Composer
   - Fields in Application Composer
   - Actions and Links in Application Composer
   - Server Scripts
   - Configure User Interface with Application Composer
   - Create a Requirements Specification Using Groovy Script
   - FAQs on Configure the User Interface with Application Composer

4 Configure Innovation Management for Integration with External Systems 57
   - Integration Tasks
   - Register Agile PLM
   - Manage Target System
   - Manage Connections
   - Map to External Systems
   - Manage Mapping to Innovation Management
   - How You Enable Item Class Mapping in Product Development Connector
   - Oracle Project Portfolio Management Cloud in Innovation Management
   - Assign Objects to Project Work Items
   - Troubleshooting

5 Appendix: Configure Agile PLM for Integration with Innovation Management 73
   - Overview on Configuring Agile PLM
   - Configure Message Protection for Agile PLM
   - Configure Users in Agile PLM
   - Innovation Management Attributes in Agile PLM (On Page Two)
   - Add Oracle Innovation Management Attributes to Required Privileges in Agile PLM
   - Configure Reference Objects in Agile PLM
   - Privileges for Innovation Management Users in Agile PLM
Preface

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

Using Oracle Applications

Using Applications Help

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

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

You can also read Using Applications Help.

Additional Resources

- **Community**: Use Oracle Cloud Customer Connect to get information from experts at Oracle, the partner community, and other users.
- **Guides and Videos**: Go to the Oracle Help Center to find guides and videos.
- **Training**: Take courses on Oracle Cloud from Oracle University.

Conventions

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

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

Documentation Accessibility

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

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

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

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

An Introduction to Innovation Management

Oracle Innovation Management and Oracle Product Development, along with Oracle Product Hub, deliver comprehensive Innovation to Commercialization capabilities across your entire product value chain.

Oracle Innovation Management consists of the following products:

- Product Requirements and Ideation Management
- Concept Design Management
- Product Lifecycle Portfolio Management

Oracle Product Development enables you to manage product data and change orders while balancing cost.

In the **Setup and Maintenance** work area, these products appear as **Functional Areas**. You can view and implement them through the **Product Management** offering.

The following table lists the functional areas and their descriptions:

<table>
<thead>
<tr>
<th>Functional Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Allows employees and stakeholders to collaborate on product innovation ideas and record requirements. Product managers can integrate requirements with concepts in Concept Design Management, and ideas with proposals in Product Lifecycle Portfolio Management.</td>
</tr>
<tr>
<td>Concept Design Management</td>
<td>Offers a collaborative design workspace for product architects, designers and executives to generate, capture, analyze, and approve product concepts that address product strategy goals. Approved concepts can then be transferred directly to Product Lifecycle Management (PLM) solutions for prototype planning, detailed design and product introduction.</td>
</tr>
<tr>
<td>Product Lifecycle Portfolio Management</td>
<td>Allows product portfolio managers to create, analyze, manage and revise product portfolios, to arrive at an optimal product mix. Portfolio managers can also optimize resources across a portfolio, evaluate portfolios, and design forecasting road maps.</td>
</tr>
<tr>
<td>Product Development</td>
<td>Uses Items, Structures (BOM), and Changes to track the development processes around products, and enable fast-track commercialization of the correct products. Product Development enables a company to incorporate concepts or early BOMs, designs, and other documents from sources such as Oracle Innovation Management or external PLM applications. PD manages changes formally and centrally on Items (parts), and Items/BOMs can be released to manufacturing with recommendations on sourcing (example, manufacturer parts).</td>
</tr>
</tbody>
</table>
For information about getting started with Oracle Cloud and implementing Oracle SCM Cloud, refer to the Oracle Cloud Documentation library.

The following table lists the chapters that describe business process setup and functional area setup tasks, in that order.

<table>
<thead>
<tr>
<th>Chapters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Innovation Management</td>
<td>Lists the setup tasks common to SCM implementation that are required prior to setting up Innovation Management. Includes information about deployment scenarios applicable to Innovation Management</td>
</tr>
<tr>
<td>Setting up the Functional Area Innovation Management</td>
<td>Provides a roadmap of setup tasks, and identifies key setup decisions that are required to set up individual modules in Innovation Management</td>
</tr>
<tr>
<td>Configuring Innovation Management</td>
<td>Provides a roadmap of setup tasks, and identifies key setup decisions that are required to set up individual modules in Innovation Management. Details the use of Data Composer and Page Composer in configuring concept rollups, and the setup of Oracle Social Network in Innovation Management.</td>
</tr>
<tr>
<td>Configuring Innovation Management for Integration with External Systems</td>
<td>Details the tasks required to configure Innovation Management for integration with external applications. Optional to implementors.</td>
</tr>
<tr>
<td>Configuring Agile PLM for Integration with Innovation Management</td>
<td>Details the tasks required to configure Agile PLM for integration with Innovation Management. Optional to implementors</td>
</tr>
</tbody>
</table>

**Related Topics**

- Innovation Management Functional Areas

**Checklist to Configure Oracle Innovation Management Cloud**

Let’s have a look at the recommended steps for implementing Oracle Innovation Management Cloud Service.

The sequence of setup tasks is split across the Cloud Service Administrator and Application Implementation Consultant roles.

The following table lists the tasks for the Service Administrator in the Cloud Customer Portal:

**Tasks for the Service Administrator in the Oracle Cloud Customer Portal**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1 - Understand Your User Profiles</strong></td>
<td>With your Oracle Applications Cloud Service, you have three user profiles:</td>
</tr>
<tr>
<td></td>
<td>• Two application profiles: used to access your Stage Environment and Production Environment.</td>
</tr>
</tbody>
</table>
Task | Description
--- | ---
| **Step 2 - Confirm Your Browser Configuration** | Confirm that you’re using one of the supported browsers with Oracle’s recommended configuration (minimum native screen resolution of 1280x1024).
  - Internet Explorer 11.x, 10.x, 9.x
  - Mozilla Firefox 24+
  - Google Chrome 35+
  - Apple Safari 7.x and 6.x

| **Step 3 - Confirm Your Application Login Credentials** | Locate your Welcome email containing your application user login credentials for the Stage and Production Environments.

**Tip:** When first provisioned, the Stage and Production Environments are assigned a default sizing for a number of concurrent users. These default values may not be adequate and may be changed to provide optimal performance. We recommend that you identify your sizing requirements early in the implementation. Ensure that they’re adjusted in advance of when you support a number of users. Supply this information through a Service Request raised with Oracle Cloud Operations who manage the environments. This helps minimize delays in your implementation.

Access the Service Administrator Action List.

| **Step 4 - Add Additional Notification Contacts** | Initially, you’re the only person at your company who receives critical Oracle notifications, including upgrade and outage schedules.

As a best practice, you can add users to receive important notifications of upgrades and outages when you’re unavailable.

1. In the Oracle Cloud portal, sign in using your Oracle.com account credentials.
2. Click the **Sign In to Notifications** button.
3. In the Oracle Notifications Portal page, click the **Users** tab.
4. Click the **Add User** button, and fill out the user information. Ensure that the Role is **Administrator**, and that you select **Yes** in the **Receive emails** field.

**Users with the role of Administrator** can add other employees to receive notifications.

| **Step 5 - Add Additional Administrators for your Oracle Applications Cloud** | We recommend appointing at least two administrators who can access My Services and perform administrative functions.

1. Locate your Welcome email and access the Service Administrator Action List.
2. Follow the **My Services** URL and login with your Oracle credentials.

**Note:** You must change your password the first time.

3. In the **Identity Domain** field, paste the Identity Domain (environment name) from your email.
4. Click **Sign In**.
5. Click **Security > Users > Add**.
6. Fill out the information for the new user, making sure to move an Administrator role to the Assigned Roles list.

**Users with a role of Administrator** can access My Account to:

- order more services

**Tip:** Check if you require to register all users, other than implementation users, as employees first. To ensure critical business functions work, create the relevant users as employees first, and then let the application create user accounts for them.
Task | Description
--- | ---
| o manage services from all identity domains and data centers for your account
| o monitor service status
| o view historical usage data
| o add Account Administrators

**Step 6 - Register Your New Customer Support Identifier in My Oracle Support**

You should have already received a separate email containing the Customer Support Identifier (CSI) for your new Oracle Cloud Service. You must register this CSI in *My Oracle Support* using your Oracle Account.

The first person to request access to a CSI is checked by Oracle to ensure that the domain of the email matches the domain associated with the CSI.

Once approved, you’re made the administrator of that CSI, and can approve access requests to your CSI.

- If this is your first time using *My Oracle Support*, you’re prompted for your CSI number after signing in. Enter your CSI number, click Request Access and follow the instructions.
- If you have previously used *My Oracle Support*, add your CSI to your *My Oracle Support* account by following these steps:
  a. After you sign in, click the More tab and select Settings.
  b. Click My Account.
  c. Click the Request Access button.
  d. In the Support Identifier field, enter your new CSI number and click the Request Access button.

  If someone else has already been made administrator for that CSI, then your request is emailed to him or her for approval.

The following table lists the tasks for the Service Administrator in the Setup and Maintenance Work Area:

**Tasks for the Service Administrator in the Setup and Maintenance Work Area**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7 - Configure the Product Management Offering</strong></td>
<td>In the Setup and Maintenance work area, scroll through the product icons and select the Product Management offering. Navigate to the Administration section. From the Actions menu, click Change Configuration.</td>
</tr>
</tbody>
</table>

Enable the following functional areas of Innovation Management for implementation:

- Product Requirements and Ideation Management
- Concept Design Management
- Product Lifecycle Portfolio Management
- Product Management Business Intelligence Analytics (if your users require BI Reports for IM)

Optionally, create Implementation Projects to assign individual setup tasks to one or more implementors.
Alternatively, click **Setup** to start the implementation process yourself.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Step 8 - Sync Oracle Innovation Management with Oracle Identity Manager (OIM)** | This task pulls information about users, roles, and roles provisioned to users, from the LDAP directory in OIM to the Oracle Cloud Applications tables.  

**Note:** Perform this task before you create implementation users so that appropriate roles are available for them.  

Search and execute the **Run User and Roles Synchronization Process** task.  

Click **Submit**. Click **OK** at the end of the process, and close the window.  

Once the Oracle Cloud Applications tables are initialized with this information, they’re maintained automatically. |

| **Step 9 - Create a Primary Implementation User** | For your consultants to access and begin your implementation process, create the primary implementation user for your lead consultant.  

**Tip:** Oracle recommends that you set up your implementation users in the Test environment first. Migrate them to Production after you test and validate them.  

On completion of the task, this user can create additional users for the rest of the implementation team.  

To create the primary user:  

1. Navigate to the Security Console.  
2. Click the Users tab.  
3. Click **Add User Account**.  
4. Enter the required information:  
   - First Name  
   - Last Name  
   - Email address  
   - User name  
   - Password  
   - Confirm Password  
5. Click **Add Role** and search for the **Supply Chain Application Administrator** role. Select and click **Add Role Membership**.  
6. Click **OK**.  
7. Search for the **Application Implementation Consultant** role. Select and click **Add Role Membership**.  
8. Click **OK**.  
9. Click **Done**.  
10. Click **Save and Close**.  

Notify your primary implementation team member that their user ID is created and give them their initial password. |

| **Step 10 - Set up Key Implementation Users and Security Profiles** | After your environments are provisioned, you as the Service Administrator have sufficient security abilities to create three implementation users with the necessary roles. |
These users are:

- **OIMAdmin**: Can access Oracle Identity Manager (OIM) to perform all required security setup functions for your implementation.
- **TechAdmin**: Can perform key technical duties, including functional setup and assigning security roles to users.
- **APPL_IMPL_CONSULTANT** and **SCM_IMPL_CONSULTANT**: Can perform key functional duties, including functional setup.

You may decide to replace or refine these initial users, but these users have all the access required to get you started.

The following table lists the tasks for the application implementation consultant in the Setup and Maintenance work area:

### Tasks for the Application Implementation Consultant in the Setup and Maintenance Work Area

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11 - Create Data Roles and Assign Security Profiles</strong></td>
<td>By default, users are denied access to all data. You can secure data by provisioning roles that provide the necessary access.</td>
</tr>
<tr>
<td></td>
<td>Data roles apply explicit data security policies on job and abstract roles. Create and maintain data roles in the Authorization Policy Manager (APM).</td>
</tr>
<tr>
<td></td>
<td>Assign a predefined security profile to relevant job or abstract roles using the Oracle Human Capital Management (HCM) setup task <strong>Manage Data Role and Security Profiles</strong>.</td>
</tr>
<tr>
<td><strong>Step 12 - Create a Legal Address and a Legal Entity</strong></td>
<td>To create application users, you must have basic HCM Corporate Structure data ready.</td>
</tr>
<tr>
<td></td>
<td>1. Search for the <strong>Manage Legal Addresses</strong> task and create a legal address.</td>
</tr>
<tr>
<td></td>
<td>2. Create a new legal entity using the <strong>Manage Legal Entity</strong> task.</td>
</tr>
<tr>
<td></td>
<td>Enable the options that identify the entity as a <strong>Payroll Statutory Unit</strong> and a Legal <strong>Employer</strong>.</td>
</tr>
<tr>
<td><strong>Step 13 - Create A Legislative Data Group and Associate it to the Legal Entity</strong></td>
<td>Use the <strong>Manage Legislative Data Groups</strong> task to create a legislative data group.</td>
</tr>
<tr>
<td></td>
<td>Use the <strong>Manage Legal Entity HCM Information</strong> task to associate the required legislative data group with the legal entity (Payroll Statutory Unit and Legal Employer).</td>
</tr>
<tr>
<td><strong>Step 14 - Create a Business Unit</strong></td>
<td>Run the <strong>Manage Business Unit</strong> task to create one or more business units.</td>
</tr>
<tr>
<td><strong>Step 15 - Create End Users</strong></td>
<td>To create application users, perform the <strong>Manage Users</strong> task in the Setup and Maintenance work area.</td>
</tr>
<tr>
<td></td>
<td>When you create a user, you must also assign the user one or more roles. Roles have all required privileges mapped to them that enable the user to perform tasks in the application.</td>
</tr>
</tbody>
</table>
Oracle Innovation Management is shipped with the following job and duty roles:

- Product Design Engineer: Concept Development Duty
- Product Design Manager: Concept Management Duty
- Product Management VP: Portfolio Management Duty
- Product Manager: Product Proposal Management Duty
- Product Portfolio Manager: Portfolio Management Duty

**Note:** You can import user data in bulk from a file.

---

### Step 16 - Perform Common Application Configuration

Common applications configuration includes setup of security, common reference objects, collaboration messaging, OTBI and configured ESS jobs, data export and import instructions, and maintenance tasks.

### Step 17 - Define Innovation Management

The following tasks per functional area are seen in the Product Management offering task list:

- **Product Requirements and Ideation Management**
  - Manage Product Idea Classes
  - Manage Product Idea Statuses
  - Manage Product Requirement Classes
  - Manage Product Requirement Statuses
  - Manage Product Requirements and Ideation Lookups

- **Concept Design Management**
  - Manage Product Concept Classes
  - Manage Product Concept Component Classes
  - Manage Product Concept Lookups
  - Manage Product Concept Statuses
  - Manage Proposal Statuses

- **Product Lifecycle Portfolio Management**
  - Manage Portfolio Statuses
  - Manage Portfolio and Product Rule Sets
  - Manage Product Portfolio Classes
  - Manage Product Portfolio Lookups
  - Manage Product Portfolio Metrics
  - Manage Product Portfolio Planning Periods

---

**Note:** Perform the **Create Implementation Users** task as an administrator to access OIM. Search for users by name, and assign required roles.
### Task Description

<table>
<thead>
<tr>
<th>Step 18 - Configure Oracle Innovation Management for Integration (Optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Oracle Innovation Management to integrate with target PLM systems or Oracle Product Development.</td>
</tr>
<tr>
<td>1. Register Agile PLM</td>
</tr>
<tr>
<td>2. Manage Target System</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 19 - Configure and Extend Oracle Innovation Management (Optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Define attributes and metrics in Innovation Management; use Application Composer to enable them.</td>
</tr>
<tr>
<td>• Enable Oracle Social Network for business objects in IM.</td>
</tr>
</tbody>
</table>

### Tasks for the PLM Administrator in External Environments

The following table describes how you can configure Agile PLM and is required only if you intend to integrate PLM with IM.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 20 - Configure a Target System for Integration (Optional)</td>
<td>Configure Agile PLM or Oracle Product Development to integrate with Oracle Innovation Management.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 20 a - Configure Agile PLM</th>
<th>The following tasks are required to integrate Oracle Innovation Management with Agile PLM only.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Configure security certificates.</td>
<td></td>
</tr>
<tr>
<td>2. Configure web service connections in Enterprise Manager.</td>
<td></td>
</tr>
<tr>
<td>4. Enable Oracle Innovation Management attributes in Agile PLM.</td>
<td></td>
</tr>
<tr>
<td>5. Add Oracle Innovation Management attributes to required privileges in Agile PLM.</td>
<td></td>
</tr>
<tr>
<td>6. Configure External References Application and Subclass in Agile PLM.</td>
<td></td>
</tr>
<tr>
<td>7. Enable required privileges for Oracle Innovation Management users to create reference objects in Agile PLM.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If you’re integrating Oracle Innovation Management to Agile PLM (9.3.4 or higher), you also require WebLogic Suite or WebLogic Suite for Oracle Applications.

### Related Topics

- User and Role Synchronization: Explained
- Creating Implementation Users: Procedure
- Creating Data Roles for Implementation Users: Procedure
- Innovation Management Functional Areas
- Configure External Systems for Integration
Deploy Innovation Management

This topic discusses deployment and integration options available to implementors of Oracle Innovation Management.

Deployment Choices

Deploy Innovation Management in Cloud environments according to your required level of control and configurability. The following table lists the cloud deployment options.

<table>
<thead>
<tr>
<th>Deployment Option</th>
<th>Deployed By</th>
<th>Level of Control and Configurability</th>
<th>Speed of Adoption and Agility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private Cloud</td>
<td>Oracle deploys and manages for you in an exclusive private Cloud.</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Oracle Public Cloud</td>
<td>Oracle provides a subscription-based service.</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

Another possible deployment scenario is of a Hybrid Cloud, where Oracle integrates Cloud deployments with legacy applications on-premise, although configurations are limited.

Integration Choices

In addition to deploying on cloud, you can integrate Innovation Management with other Oracle Cloud and external PLM applications.

- **Oracle Product Development Cloud**
  Streamline new product development and introduction processes. Innovation Management Cloud with Product Development Cloud helps in rapidly innovating and developing the best mix of profitable products.

- **Oracle Project Portfolio Management Cloud**
  Track the conversion of ideas to projects and profitable products. Use tasks associated with work items for requirements specification, concept, and proposals of the project.

- **Oracle Agile Product Lifecycle Management**
  Leverage legacy items and PLM processes by integrating Agile PLM with Innovation Management (in a Hybrid Cloud).
2 Set Up Innovation Management Functional Areas

Innovation Management Functional Areas

This topic outlines the default tasks required to define Innovation Management in the Setup and Maintenance work area. You must first complete the common application setup and configuration tasks for Product Management.

In the Setup and Maintenance menu, click the Product Management offering. Navigate to the required task list to view the functional area tasks.

The table lists functional areas and tasks for Innovation Management.

<table>
<thead>
<tr>
<th>Task List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Product Requirements and Ideation Management</td>
<td>Use this task list to configure ideas, and requirements specifications.</td>
</tr>
<tr>
<td>• Manage Product Idea Classes</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Idea Statuses</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Requirement Classes</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Requirement Statuses</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Requirements and Ideation Lookups</td>
<td></td>
</tr>
<tr>
<td>Define Concept Design Management</td>
<td>Use this task list to configure concepts and concept components.</td>
</tr>
<tr>
<td>• Manage Product Concept Classes</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Concept Component Classes</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Concept Statuses</td>
<td></td>
</tr>
<tr>
<td>• Manage Proposal Statuses</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Concept Lookups</td>
<td></td>
</tr>
<tr>
<td>Define Product Lifecycle Portfolio Management</td>
<td>Use this task list to configure proposals and portfolios.</td>
</tr>
<tr>
<td>• Manage Product Portfolio Classes</td>
<td></td>
</tr>
<tr>
<td>• Manage Portfolio Statuses</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Portfolio Planning Periods</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Portfolio Metrics</td>
<td></td>
</tr>
<tr>
<td>• Manage Product Portfolio Lookups</td>
<td></td>
</tr>
</tbody>
</table>
Configure External Systems for Integration

Oracle Innovation Management integrates with systems such as Agile PLM, and Oracle Product Development, through view objects, application modules, and web services.

You can perform the following tasks by integrating target systems:

- **Associate issues to ideas**
  - Relate issues to concepts that improve products
  - Relate issues or ideas to requirements to drive improved designs

- **Create items in Product Development from concept components**
  - Search and use an existing item to create a concept
  - Associate requirements to Agile items (trace which requirements were used to build the eventual product)
  - Map attributes from Agile PLM items to concept components

- **Allow proposals to be related to or drive projects in Agile PPM**
  - View status and key attributes of Projects within proposals
  - Integrate actual cost and resources from Agile PLM Project into proposals
  - Check how a proposal is progressing against projected cost and resources

**Related Topics**

- Integration Tasks
- Overview on Configuring Agile PLM

Pages and Page Layouts in Application Composer

Use the Pages node to create and modify page layouts. Page layouts define what fields users will see and modify. A page layout is a design of a page which you create, and then attach conditions to. Using conditions, you can present the same Oracle Cloud Innovation Management page differently to various users. For example, a product manager can see one version of the page, while the sales representative sees another version. Page layouts are available only for simplified pages.

To create a new layout:

- Click the Pages node in the desired object you want to modify.
- Click the Duplicate icon for the type of layout you intend to define, such as the Details Page layout.
- Name the layout, and click OK.
- In the Summary area, click the Edit icon.
In the **Available Fields** panel, select the fields that you want to display and move them to the **Selected Fields** panel.

### Working with Page Layouts for Standard Objects

Standard objects that have simplified pages are delivered with default page layouts, called standard layouts. Standard layouts are the pristine model layouts that you can’t edit. However, you can duplicate the standard layout to create a new layout. You can edit configured layouts and add display conditions to them. When a configured layout is no longer of use, you can inactivate, or deprecate, that layout.

You can:

- **Duplicate page layouts**
  
  To create a new page layout, duplicate an existing layout and then make your edits.
  
  All supported objects are delivered with a standard layout for their simplified user interface pages. Duplicate the standard layout to configure new layouts, which you can edit.
  
  The first configured layout for a page type is automatically named the default configured layout, but you can change the name.

- **Edit configured layouts**
  
  You can edit only configured layouts. Configured layouts are duplicated from an existing layout.

- **Inactivate, or deprecate, configured layouts**
  
  You can delete a page layout from a duplicated sandbox that has not been published.
  
  You cannot delete a page layout from a published sandbox, but you can inactivate configured layouts by deselecting the Active check box for a page layout on the Simplified Pages tab.
  
  You cannot inactivate the standard layouts that are automatically delivered for an object.

### Working with Page Layouts for Configured Objects

Working with page layouts for configured objects is exactly the same as working with page layouts for standard objects. The only difference is that after you create a configured object, you must manually create its set of configured layouts before you can start to work with them. Configured objects do not have a set of standard layouts.

Create a set of simplified page layouts for a configured object with a click of a button. Clicking that button tells Application Composer to automatically create the following:

- Page layouts for the object’s set of user interface pages, such as the creation and details pages.
- A Search and Select dialog, which you can configure by clicking the Edit Picker link.

### Standard and Configured Layouts

Standard layouts are the pristine model layouts that you can’t edit. Configured layouts are copies of standard layouts that you make, which you can edit.

Standard layouts exist to make your upgrades seamless. When you upgrade to a new release of Oracle Cloud Innovation Management, Oracle upgrades only the standard layouts for each object. Your configured layouts are not touched. This makes it easy for your users to continue working immediately after an upgrade. In the meantime, you can take your time to
review the changes that happened to standard layouts as part of an upgrade, and manually incorporate those changes as and when needed.

Related Topics
- Capabilities of Application Composer per Object Type
- Fields in Application Composer

Class Management in Oracle Innovation Management

Class Management is the definition of classes, class hierarchies, and class codes to establish reusable business objects. Here we learn about class management for ideas, requirements specifications, concepts, and portfolios in Oracle Innovation Management.

The tasks addressed here are:
- Manage Product Idea Classes
- Manage Product Requirement Classes
- Manage Product Concept Classes
- Manage Product Concept Component Classes
- Manage Product Portfolio Classes

Class

Use classes and subclasses to define business object types.

When you create a class, the class name that you provide is stored and used as an object type, at the time of business object creation.

Select a class to edit the class name and description. The Object Creation Allowed Indicator in the Edit Class page controls the possibility of creating business objects of the current class value. Select the indicator to ensure that the class name is available to use as a type when creating a business object.

Class Code

A Class Code is a constant and unique value associated with each class across Oracle Innovation Management and associated PLM systems.

You can define a class code only once, when creating a class, as it’s used during integration with external systems, and is required to remain a consistent internal code.

Note: You can’t edit the class code after class creation. However, you can delete the existing class, if it isn’t already used to create an object, and create a class with the required class code.

Class Hierarchy

Class Hierarchy enables you to group and search for classes, based on class values or business objects types.
Select a class in the Manage Class page to view the class hierarchy in the Edit Class page.

**Innovation Management Lookups**

Oracle Innovation Management provides lookups that you can use to define values in Requirements, Concept, and Portfolio modules during implementation.

This topic addresses the following tasks:

- Manage Product Requirements and Ideation Lookups
- Manage Product Concept Lookups
- Manage Product Portfolio Lookups

Use standard lookups in Oracle Innovation Management to define values such as type, status, priority, scope, compliance, resource pool, metrics, lifecycle phases, and rank.

Application statuses are also standard lookups. You can execute the following tasks as lookup tasks:

- Manage Product Idea Status
- Manage Product Requirement Status
- Manage Product Concept Status
- Manage Proposal Status
- Manage Portfolio Status

The following table details the standard lookups available in Oracle Innovation Management.

Lookups with configuration level System do not allow you to add or delete lookup codes. However, you can edit the **Meaning** and **Description** fields of the existing lookup codes.

<table>
<thead>
<tr>
<th>Application</th>
<th>Module</th>
<th>Lookup Type</th>
<th>Lookup Code Meaning</th>
<th>Configuration Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Ideas</td>
<td>Status</td>
<td>Pending, Accepted, Rejected, Implemented, In Progress, Review</td>
<td>User</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Associated Product</td>
<td>Common Services, Product Concept Design, Product Lifecycle Portfolio Management, Product Requirements and Ideation Management</td>
<td>User</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Fulfillment</td>
<td>Yes, No</td>
<td>User</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Priority</td>
<td>Must Have, Nice to Have, Should Have</td>
<td>User</td>
</tr>
<tr>
<td>Application</td>
<td>Module</td>
<td>Lookup Type</td>
<td>Lookup Code Meaning</td>
<td>Configuration Level</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-------------------------</td>
<td>----------------------</td>
<td>-------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Scope</td>
<td>Yes, No</td>
<td>System</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Requirement Status</td>
<td>Pending, Submitted, Released</td>
<td>System</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Requirements</td>
<td>Comment Status</td>
<td>Open, Closed</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Proposal Business Unit</td>
<td>Business Unit</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Concept Product Type</td>
<td>New Product, Technology Evaluation, Product Redesign</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Component Product Type</td>
<td>Documentation, Electrical, Mechanical, Software, Tooling</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Concept Status</td>
<td>Draft, Submitted Approved, Converted</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Cost Category</td>
<td>Development, Production</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Cost Status</td>
<td>Actual, Projected</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Proposal Cost Types</td>
<td>Fixed, Labor, Material, Variable</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Funding Request For</td>
<td>Concept, Feasibility, Product, Prototyping</td>
<td>User</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Market Strategy</td>
<td>Differentiation, Neutralization, Optimization, Others</td>
<td>System</td>
</tr>
<tr>
<td>Product Concept Design</td>
<td>Concepts</td>
<td>Primary Justification</td>
<td>Enter New Markets, Enter New Regions, Exit Market, Expand Market Share</td>
<td>User</td>
</tr>
<tr>
<td>Product Lifecycle Portfolio Management</td>
<td>Portfolio</td>
<td>Product Maturity Chart Metrics</td>
<td>Cost, Revenue, Margin</td>
<td>User</td>
</tr>
<tr>
<td>Product Lifecycle Portfolio Management</td>
<td>Portfolio</td>
<td>Portfolio Metric Data Types</td>
<td>Cost, Number, List of Values</td>
<td>System</td>
</tr>
</tbody>
</table>
### Application | Module | Lookup Type | Lookup Code Meaning | Configuration Level
--- | --- | --- | --- | ---
Product Lifecycle Portfolio Management | Portfolio | Metric Types | Product, Portfolio, Product Proposal | System
Product Lifecycle Portfolio Management | Portfolio | Portfolio Planning Period Statuses | Active, Inactive | System
Product Lifecycle Portfolio Management | Portfolio | Planning Period Unit Durations | Month, Quarter | System
Product Lifecycle Portfolio Management | Portfolio | Lifecycle Phases | Draft, Submitted, Approved, Released | System
Product Lifecycle Portfolio Management | Portfolio | Portfolio Metrics Rank | 1,2,3,4,5,6 | User
Product Lifecycle Portfolio Management | Portfolio | Portfolio Types | Regular, Template | System
Product Lifecycle Portfolio Management | Portfolio | Product Categorization Metric | Cash Cow, Dog, Flagship, Star, Question Mark | User
Product Lifecycle Portfolio Management | Portfolio | Product Metrics Rank | 1,2,3,4,5,6 | User
Product Lifecycle Portfolio Management | Portfolio | Product Risk Numeric | Low, Medium, High, No Risk | User
Product Lifecycle Portfolio Management | Portfolio | Product Risk Subjective | No Risk, High, Medium, Low | User
Product Lifecycle Portfolio Management | Portfolio | Portfolio Risk Numeric | Low, Medium, High | User
Product Lifecycle Portfolio Management | Portfolio | Portfolio Risk Subjective | Low, High, Medium | User
Product Lifecycle Portfolio Management | Portfolio | Scenario Types | Draft, Submitted, Approved, Rejected, Current | System

### Related Topics
- Overview of Lookups
- How can I edit lookups
- Example of a Standard Lookup
Configure Planning Periods for Your Portfolio

Product portfolio planning period is the time period during which the portfolio objects collect data for analysis. Before defining a planning period, create planning period units using the Manage Product Portfolio Planning Periods task in the Setup and Maintenance work area.

Provide start date, number of units and specify the duration, which can be either monthly or quarterly, to create planning period time units. The planning period time units are created with default labels. You can add time units for a previously created planning period unit by providing the number of units before the first unit or by providing the number of units after the last unit.

Note: After you create a planning period unit specifying a duration, you can’t change the duration. However, you can change names of the units.

To create a planning period, provide the start date planning period unit, end date planning period unit, and select the planning period unit from the choice list.

Impact of planning period and planning period units:

- The columns in the Manage Resource Capacity table is dynamically created and displayed based on the number of units defined in the planning period for the portfolio.
- Each timeline in the Schedule, Resource, and Launch charts is determined by the number of units defined in the planning period for the portfolio.

Portfolio Metrics

Oracle Innovation Management offers a list of predefined metrics that you can use for measuring portfolio performance. You can also set the minimum and maximum threshold values for these metrics in accordance with business requirements.

Metrics marked as Enabled are made available for selection when you add metrics using Actions > Define Metrics. Enabled metrics that are also marked as Default appear in the locations described in the following table.

<table>
<thead>
<tr>
<th>Default metrics for:</th>
<th>Appear here:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product proposals</td>
<td>In the Proposal Metrics selection dialog that opens when you click the 123 icon during proposal creation.</td>
</tr>
<tr>
<td>Products</td>
<td>On Edit Portfolio page &gt; Actions &gt; Define Metrics leads to Define Metrics page, with Product Metrics tab.</td>
</tr>
<tr>
<td>Portfolios</td>
<td>On Edit Portfolio page &gt; Actions &gt; Define Metrics leads to Define Metrics page, with Product Portfolios tab.</td>
</tr>
</tbody>
</table>
Note: A portfolio is a grouping of product lines, most often within company Business Units. All proposals related to product lines within a portfolio should ideally be evaluated by a common set of metrics so that the evaluations are consistent and objective. If you define consistent metrics for a portfolio and proposals within that portfolio, you can then roll up portfolio metrics from proposals, as required.

Metrics are of three types:

- **Derived** - Calculated using fixed formulas. (Examples: Return On Investment, Internal Rate of Return)
- **Derived and rolled up** - Calculated using fixed formulas and derived from certain values that you enter. (Examples: Net Present Value, Actual Cost, Projected Cost).
- **User entered** - Entered by the user in the user interface. (Examples: Impact, Alignment, Risk Numeric)

**Related Topics**

- How You Calculate Net Present Value
- How You Calculate Internal Rate of Return
- How You Calculate Break Even Time
- How You Calculate Payback Period
- Checklist to Configure Oracle Innovation Management Cloud

**Set Up Concept Rollups**

With the required administrator privileges, you can create concept metrics and configure how these are rolled up. Product managers can evaluate, approve or reject product concepts against these metrics. You can also rename existing concept metrics using Application Composer.

To set up concept rollups:

- Navigate to the Concept Design work area.
- On the Manage Concepts page, expand the Tasks panel and click Setup Concept Rollups.
- On the Concept Rollups page, select the fields to use for the rollup.

The following table lists administrator-defined metrics that you can configure.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>If selected, includes the rollup definition in the calculation.</td>
</tr>
<tr>
<td>Rollup Column</td>
<td>Column of attributes selected for rollup and aggregation.</td>
</tr>
<tr>
<td>Result Column</td>
<td>Destination column that displays rolled up results and appears in the General Information tab of each component to all users.</td>
</tr>
<tr>
<td>Consider Quantity</td>
<td>If selected, multiplies the attribute in the Rollup column with the quantity in the Quantity column of the component.</td>
</tr>
</tbody>
</table>
Set Up Innovation Management Functional Areas

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete Rollup Count Column</td>
<td>Column that displays the number of leaf nodes that do not have a value entered in the rollup column.</td>
</tr>
<tr>
<td>Target Value Column</td>
<td>Identifies the column of data to compare with the Rollup column to determine a variance. The default calculation for variance is: Variance = Rollup Result Column - Target Value Column</td>
</tr>
<tr>
<td>Variance Result Column</td>
<td>Destination column that displays the calculated variance.</td>
</tr>
</tbody>
</table>

**Related Topics**

- How You Calculate Metrics in Concepts
- Analyze a Product Concept
- Rollups in Concept Design
- Fields in Application Composer

**Portfolio and Product Rule Sets**

Define portfolio and product rule sets to associate multiple rules together, and assign them to portfolio classes. The following table lists and describes the portfolio and product rule sets.

<table>
<thead>
<tr>
<th>Type of Rule Set and Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>Define validation conditions based on attribute values.</td>
</tr>
<tr>
<td></td>
<td>predefined business rules</td>
</tr>
<tr>
<td></td>
<td>Logical expression</td>
</tr>
<tr>
<td></td>
<td>Validation condition</td>
</tr>
<tr>
<td></td>
<td>User message</td>
</tr>
<tr>
<td>Assignments</td>
<td>Define the value of an attribute, based on the specified condition. Rules are executed in the order of their sequence in the rule set.</td>
</tr>
<tr>
<td></td>
<td>Target business entities are:</td>
</tr>
<tr>
<td></td>
<td>Portfolio General Information</td>
</tr>
<tr>
<td></td>
<td>Product</td>
</tr>
<tr>
<td></td>
<td>Scenario</td>
</tr>
<tr>
<td>Composite</td>
<td>Aggregate rules sets that operate on different attribute groups. Composite rule sets contain both validation and assignment rule sets.</td>
</tr>
</tbody>
</table>
Enable Single Sign-On for Innovation Management

You can make it possible for your users to use a single user name and password to sign in to all of your on-premises and Oracle SCM Cloud applications by implementing Oracle Enterprise Single Sign-On.

Oracle Enterprise Single Sign-On (SSO) is available for customers who have implemented either the Microsoft Active Directory Federation Server 2.0 or the Oracle Identity Federation Service 11g identity provider (IdP). Other identity providers require special approval. Here is the standard approval process:

1. Contact your salesperson or open a service request for SSO Enablement on support.oracle.com
2. Your Oracle sales or help desk sends you a questionnaire to fill out.
3. After you return the questionnaire, Oracle representatives evaluate your responses and obtain approval, usually within 24 hours.
4. After you are approved, Oracle sets up your Oracle SCM Cloud environment and you receive the appropriate documentation on setting up your system.

After you obtain necessary approval, it takes a minimum of two weeks for Oracle to implement the SSO for your Oracle SCM Cloud environment.

FAQs on Social Networking in Oracle Innovation Management

What are the prerequisites for Oracle Social Network integration?

For Oracle Social Network integration to work, there must be a connection defined between Oracle Fusion applications and Oracle Social Network. The administrator must configure this connection, as Oracle Fusion applications and Oracle Social Network do not reside on the same server.

Ensure that the following configuration items are in place:

- Oracle Social Network URL must be defined in Topology Manager with a Module Short Name.
- The Oracle Social Network credential must exist in the WebLogic Server credential store.

Related Topics

- Overview of the Oracle Social Network Objects in Oracle Innovation Management
How do I share my opinion with development teams on terminating a product?

If the Edit Portfolio page has a Social link, you can invite others to a conversation to discuss the ideas.

For example, as a portfolio manager, you carefully weigh the market share research, revenue, and future portfolio plans, and suspect that it may be time to end support for one of the products in your portfolio. You want to make sure you have agreement from the people closest to the product, the product development manager, and the product manager.

From the Edit Portfolio page:

1. Click the Social link to open Oracle Social Network (OSN).
2. Click New Conversation.
3. Invite your product manager and product development manager to the conversation.

The details of your conversation and key aspects of the portfolio are visible on the portfolio wall in OSN for everyone to view.

After a joint online discussion about the pros and cons of terminating the product, questions are asked and answered, and supporting documents are uploaded and reviewed. When you click the Social link from a business object, all the social networking features provided by OSN are instantly available. This makes it easy to bring in the people you require to make an informed decision.

Depending on your job role and permissions, you can use social networking features for the following Oracle Innovation Management business objects:

- Ideas
- Requirements Specifications
- Concepts
- Proposals
- Portfolios

Related Topics
- Management of Oracle Social Network Objects

How do I share my ideas with key stakeholders in Oracle Innovation Management?

If the Manage Ideas page has a Social link, you can invite others to a conversation to discuss the ideas.

For example, while at a customer meeting, an idea for enhancing the product emerges, that you think is worth pursuing. You want to get it into the application so that the product manager and other stakeholders can consider it as they weigh options for the next release.

From the Manage Ideas page:

1. Add the details of product enhancement in the form of an idea, and post it.
2. Click the Social link to open Oracle Social Network.
3. Click New Conversation.
4. Invite your product manager and product development manager to the conversation.

The details of your conversation and key aspects of the Idea are visible on the Idea wall in Oracle Social Network for everyone to view.

You might decide to share customer views about the idea, and post the customer e-mail ID to the Conversation in the form of a document.

After several days of discussion on the Idea, you collectively decide that the idea has enough merit to move forward. The product manager creates a Requirement to get the ball rolling, frequently referring back to the Conversation to see what was said. The Idea is thus translated into tangible requirements that can be measured in reality.

Depending on your job role and permissions, you can access social networking features for the following Oracle Innovation Management business objects:

- Ideas
- Requirements Specifications
- Concepts
- Proposals
- Portfolios

Related Topics
- Management of Oracle Social Network Objects
3 Configure the User Interface with Application Composer

Overview of the Application Composer for Oracle PLM Cloud

Application Composer is a browser-based tool that an administrator can use to configure applications. Use this tool to make data model changes that previously required application developers. Administrators can create and configure layouts to meet business requirements.

For example, create a new object and related fields and then create new interface pages to expose that object to users. Application Composer is a design-at-runtime tool, which means that you can navigate to Application Composer directly from a Cloud application, make your changes, and see most changes take immediate effect, without having to sign back into the application.

Roles and Privilege to work in Application Composer

An implementation user must be assigned these roles in order to access and work in Application Composer:

1. Application Implementation Consultant role, or any role that contains the Manage Extensible Object privilege.
2. Custom Objects Administration role.

Basic Capabilities of Application Composer

Use Application Composer to:

- Edit the display label and help text of standard fields;
- Create conditional layouts;
- Assign fields to layouts;
- Create fields of different types (such as text, number, date, choice list, and check box) and add them to standard and administrator-defined objects;
- Define application actions using validation rules, triggers, and functions;
- Set field-level and object-level validation rules.

Attributes, or fields, must be assigned to a layout in order for the application user to see and work with them. A conditional statement assigned to a layout determines when it is displayed and who can see it.

Note: Application Composer replaces such configuration tools as Data Composer and Page Composer. Previously created objects, attributes and other configured entities are all carried over when you upgrade your Oracle PLM Cloud applications. However, administrator-configured entities are not initially visible to the user. Previously configured attributes become visible again when they are added to a layout.

Note: Application Composer is supported for use only in English. Additionally, Application Composer is not supported for use with iPad devices.

The Application Composer enhancements enable you to create actions, buttons and URL tabs on issues and actions and import issue and action configurations from one environment to another. This enables you, an administrator, to configure
user interface and business logic to better suit user needs. They trigger administrator-defined actions and let you link and display administrator-defined objects as side tabs. Using Application Composer, configure the **Actions** menu to hide or reorder standard and administration-created actions based on their usage. Set conditions on page layouts that only users with specific privileges can access certain actions.

## How to Upgrade to an Oracle PLM Cloud Release that Features Application Composer

When you upgrade, ensure that you recreate interface configurations, which were previously modified using the Data Composer or Page Composer tools with the Application Composer tool. These include added and reordered attributes.

The following table lists configurations and instructions on how to resolve or re-configure the same interface entities with Application Composer.

<table>
<thead>
<tr>
<th>Earlier configurations with Page Composer or Data Composer</th>
<th>Resolution or Re-configuration with Application Composer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previously configured attributes that were modified or added into your application’s object pages</td>
<td>Use Page Layouts in Application Composer to make previously configured attributes visible again.</td>
</tr>
<tr>
<td>Attribute placement that included reordering with Page Composer of standard and previously configured attributes</td>
<td>Use Page Layouts in Application Composer to reorder standard and configured attributes or fields.</td>
</tr>
<tr>
<td>Entities that were previously added to your applications, for example:</td>
<td>Use Page Layouts in Application Composer to reorder and reconfigure your user interface scheme.</td>
</tr>
<tr>
<td>• Images</td>
<td>• Images are supported in Application Composer, but you must bring an image from your earlier configurations as an Attachment.</td>
</tr>
<tr>
<td>• Hyperlinks and Web pages</td>
<td>• URLs and embedded Web pages are supported in Application Composer, but you must bring them from your earlier configurations using Custom Objects.</td>
</tr>
<tr>
<td>• Boxes</td>
<td>• Boxes are supported in Application Composer, but you must bring a box from your earlier configurations as a Field Group.</td>
</tr>
<tr>
<td>• Text or HTML markup</td>
<td>• Headers in Text or HTML aren’t supported in Application Composer.</td>
</tr>
<tr>
<td>Show / Hide attributes and Field label changes using Expression Language</td>
<td>Use Groovy scripts in Application Composer for Show / Hide attributes and Field label changes.</td>
</tr>
<tr>
<td>Page Composer-driven label and font changes, or any CSS changes</td>
<td>These are lost in upgrading: Application Composer doesn’t support font or CSS changes.</td>
</tr>
</tbody>
</table>

*Note:* Other forms of data and content that were entered into your applications in earlier releases should be available after upgrading your installation of Oracle PLM Cloud. This section and table applies to the possible impact of elements that were configured in Page Composer or Data Composer in previous releases.
Capabilities of Application Composer per Object Type

Use Application Composer to create attributes and configure the listed Innovation Management or Quality Management object types.

**Note:** In Oracle PLM Cloud applications, you are always in a live environment. When you plan to create user interface entities in Application Composer, you must first open a practice state. Once you open a sandbox, it is safe to open Application Composer and work without changing the production interface pages until you are completely prepared.

Object Types that you can configure on Page Layouts

The following table summarizes the object types that you can configure on specific page layouts in Innovation Management and Enterprise Quality Management with Application Composer. The object types listed here are available on the Application Composer user interface.

<table>
<thead>
<tr>
<th>LANDING PAGE</th>
<th>CREATE DIALOG</th>
<th>DETAILS PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

**Note:** A Create dialog must include all fields that are flagged as Required. This includes Required fields that you have renamed. If not, an error message appears as you set up the Create dialog for a business object.

<table>
<thead>
<tr>
<th>Idea</th>
<th>Idea</th>
<th>Idea</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Specification and Requirements</td>
<td>Requirements Specification and Requirements</td>
<td>Requirements Specification and Requirements</td>
</tr>
<tr>
<td>Proposal</td>
<td>Proposal</td>
<td>Proposal</td>
</tr>
<tr>
<td>Concept</td>
<td>Concept</td>
<td>Concept</td>
</tr>
</tbody>
</table>

**Note:** Concepts cannot be reconfigured on the Create dialog.

<table>
<thead>
<tr>
<th>Portfolio</th>
<th>Portfolio</th>
<th>Portfolio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality Action</td>
<td>Quality Action</td>
<td>Quality Action</td>
</tr>
<tr>
<td>Quality Issue</td>
<td>Quality Issue</td>
<td>Quality Issue</td>
</tr>
</tbody>
</table>
Kinds of Configuration for each Object Type

This table summarizes the specific kinds of configuration you can perform on the object types, which are available to Application Composer and were named in the previous table.

<table>
<thead>
<tr>
<th>KINDS OF CONFIGURATION</th>
<th>APPLIES TO THESE OBJECT TYPES</th>
<th>DOES NOT APPLY TO THESE OBJECT TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Attribute Types, which include:</td>
<td>Innovation Management:</td>
<td>Innovation Management:</td>
</tr>
<tr>
<td>• Dynamic Choice List</td>
<td>• Idea</td>
<td>• Long text is not supported in Concept structures.</td>
</tr>
<tr>
<td>• Check box</td>
<td>• Requirements Specification and Requirement</td>
<td></td>
</tr>
<tr>
<td>• Percentage</td>
<td>• Proposal</td>
<td></td>
</tr>
<tr>
<td>• Date and Time</td>
<td>• Concept and Component</td>
<td></td>
</tr>
<tr>
<td>• Long text</td>
<td>• Portfolio</td>
<td></td>
</tr>
<tr>
<td>Quality Management:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality Issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page Layouts</td>
<td>Innovation Management:</td>
<td>Not applicable.</td>
</tr>
<tr>
<td>• Idea</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Requirements Specification and Requirement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Proposal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Concept and Component</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Portfolio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Management:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality Issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show or Hide Tabs</td>
<td>Innovation Management:</td>
<td>Innovation Management:</td>
</tr>
<tr>
<td>• Idea</td>
<td>• Requirement</td>
<td></td>
</tr>
<tr>
<td>• Requirements Specification</td>
<td>• Component</td>
<td></td>
</tr>
<tr>
<td>• Proposal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Concept</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Portfolio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Management:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality Issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Quality Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configure Buttons or Actions</td>
<td>Innovation Management:</td>
<td>Innovation Management:</td>
</tr>
<tr>
<td>• Idea</td>
<td>• Requirement</td>
<td></td>
</tr>
<tr>
<td>• Requirements Specification</td>
<td>• Concept</td>
<td></td>
</tr>
<tr>
<td>• Proposal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KINDS OF CONFIGURATION</td>
<td>APPLIES TO THESE OBJECT TYPES</td>
<td>DOES NOT APPLY TO THESE OBJECT TYPES</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Portfolio</td>
<td></td>
</tr>
</tbody>
</table>

**Quality Management:**
- Quality Issue
- Quality Action

<table>
<thead>
<tr>
<th>Field Groups</th>
<th>Innovation Management:</th>
<th>Not applicable.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Idea</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Requirements Specification and Requirement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Proposal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Concept and Component</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Portfolio</td>
<td></td>
</tr>
</tbody>
</table>

**Quality Management:**
- Quality Issue
- Quality Action

<table>
<thead>
<tr>
<th>Configure URL Tabs</th>
<th>Innovation Management:</th>
<th>Innovation Management:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Idea</td>
<td>Requirement</td>
</tr>
<tr>
<td></td>
<td>Requirements Specification and Requirement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Proposal</td>
<td>Quality Management:</td>
</tr>
<tr>
<td></td>
<td>Concept and Component</td>
<td>• Quality Issue</td>
</tr>
<tr>
<td></td>
<td>Portfolio</td>
<td>• Quality Action</td>
</tr>
</tbody>
</table>

**Quality Management:**
- Quality Issue
- Quality Action

<table>
<thead>
<tr>
<th>Configure first-level Objects</th>
<th>Innovation Management:</th>
<th>Innovation Management:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Idea</td>
<td>Requirement</td>
</tr>
<tr>
<td></td>
<td>Requirements Specification and Requirement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Proposal</td>
<td>Quality Management:</td>
</tr>
<tr>
<td></td>
<td>Concept and Component</td>
<td>• Quality Issue</td>
</tr>
<tr>
<td></td>
<td>Portfolio</td>
<td>• Quality Action</td>
</tr>
</tbody>
</table>

**Quality Management:**
- Quality Issue
- Quality Action

<table>
<thead>
<tr>
<th>Configure Child Objects</th>
<th>Innovation Management:</th>
<th>Innovation Management:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ideas</td>
<td>Requirement</td>
</tr>
<tr>
<td></td>
<td>Requirements Specification and Requirement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Proposal</td>
<td>Quality Management:</td>
</tr>
<tr>
<td></td>
<td>Concept and Component</td>
<td>• Quality Issue</td>
</tr>
<tr>
<td></td>
<td>Portfolio</td>
<td>• Quality Action</td>
</tr>
</tbody>
</table>

**Innovation Management:**
- Idea
- Requirements Specification and Requirement
- Proposal
- Concept and Component
- Portfolio

**Quality Management:**
- Quality Issue
- Quality Action
Getting Started in Application Composer

The following procedure is an extremely abbreviated sequence to open Application Composer and have a look around.

1. Navigate to the Application Composer work area.

   Note: In Oracle Cloud applications, you are always in a live environment. When you plan to create user interface entities in Application Composer, first open a practice state, called a sandbox, usually through Settings and Actions > Manage Sandboxes. Once a sandbox is open, it is safe to open Application Composer and work without changing the production interface pages until you are completely prepared.

2. When you open Application Composer, the Application list offers CRM Cloud and ERP and SCM Cloud environments. Select ERP and SCM Cloud.

3. In the Objects row are object tags. Innovation and Quality are out-of-the-box, configurable object tags. Your company produces object comprised by one or two of the object tags, and these are the only one that you have access to. (See the Related Link about Capabilities.) A series of business objects are now displayed.

4. In this brief procedure, you can choose a business object, and we are selecting Idea. An Idea is a standard object, populated with fields, or attributes. In the Fields node, try creating a custom field, a Text box. The path is: Objects > Standard Objects > Idea > Fields > Create > Select Field Type.

5. After creating the fields, go to Pages, assign the Text fields to a duplicated Landing page layout, and click Save.

6. Open Ideas, that is, Tasks > Manage Ideas to see the new attribute.

Example of How You Work with Sandboxes in Application Composer

When configuring business objects in Oracle PLM Cloud, use the Sandbox Manager to work within sandboxes, which are separate areas that users can’t see. In the global region, expand the Settings and Actions menu that is available when you click your user name. Select Administration > Manage Sandboxes.

Because you are always in a live environment in Oracle Cloud applications when you plan to work in Application Composer to create user interface entities, you must first open a practice state, called a sandbox. Once in a sandbox, it is safe to open Application Composer and work without changing the production interface pages.

Within this topic, the words configure and configuring refer to any modification to an existing artifact, for example, adding a new field to an existing business object. Configuring also refers to changing or adding to what is displayed on a page, as well as creating a completely new artifact, such as a business object or page.
Note: It is recommended to always use sandboxes and not to make your configurations directly in the mainline code - even when you use the Setup and Maintenance work area.

Sandbox Usage

Sandboxes typically have one of two purposes:

- **Test-Only sandbox**: Users perform all configurations using the Test-only sandbox. Changes made in the test-only sandbox must never be published to the mainline code.

- **Integration sandbox**: Once satisfied with the configurations made in the test-only sandbox, users replicate their changes in this sandbox, and then publish them to the mainline code. The Integration sandbox type is called the **Publish sandbox**, because teams working in parallel use this sandbox as the final staging point before publication to the mainline code.

Note: Prior to upgrade, users must either publish open sandboxes or delete them. After the upgrade they can create new sandboxes and publish them.

Using the Sandbox Manager, you can perform these tasks:

1. Create a sandbox.
2. Activate a sandbox.
3. Delete a sandbox.
4. Publish a sandbox.
5. View available or published sandboxes.

Creating a Sandbox

Follow these steps to create a sandbox.

1. Click your user image or name in the global header, then within Administration select Manage Sandboxes. The Manage Sandboxes window appears, listing the available sandboxes in your environment.
2. Click the New (plus sign) icon.
3. In the Create Sandbox window, enter a name in the Sandbox Name field, then click Save and Close.
4. To activate the sandbox, select the sandbox that you just created, and click the Set as Active button.

Whenever you are signed in to the application and working in a session sandbox, you can see the session sandbox name. You can work in the sandbox and perform typical extension functions. Other users can’t see what you have done until you publish your sandbox. You get to see more details of your sandbox by hovering over the sandbox name.

Note: If you sign out and sign in again as the same user, you will still be in the same sandbox. The sandbox you’re working in is a part of your user profile information.

5. Click your sandbox name in the window. As you make extensions in your sandbox, various XML files in the MDS repository are changed. In this case, we’re taking a quick tour of sandboxes, and have not made any changes, so there aren’t any files showing. If you make changes while in your sandbox, this is a way to see what all those changes are, exactly which XML files in the MDS repository have been changed, and the layers of those changes.
6. Click the More link. The Sandbox Details window appears.
7. Click the Close button.
   To exit the sandbox click the sandbox name and click Exit Sandbox.

Note: You can delete sandboxes, but remember that you can delete only those which are not published. Before you delete a sandbox confirm that the sandbox is not active. Deletion of partial content of a sandbox is risky; it is recommended that you do not use this option. After you have tested your application changes, move those changes to the integration sandbox. Publish your integration sandbox and then delete all the test-only sandboxes. You can then create and work in new sandboxes, including a new integration sandbox.

Note: The web service in the active sandbox takes precedence over the web service from the mainline (published sandbox).

Customizations Affect Metadata Services and the Database Layer

At a technical level, your configurations affect two major areas: the Metadata Services (MDS) repository and the database layer.

1. All changes result in the creation or updating of many files within the MDS repository. Your configurations are stored as XML files in the repository, organized by sandbox.
2. As custom objects and fields are created, their definitions are allocated to generic placeholders that already exist as tables or columns in the database.
3. Sandboxes handle metadata configurations made to the data stored in the Metadata Services (MDS) repository.

Related Topics
- Pages and Page Layouts in Application Composer
- Overview of Sandboxes

Layout Nodes in Application Composer

Oracle Innovation Management Cloud and Quality Management objects can be configured through Application Composer. When you open Application Composer, each object type that is available to you is displayed, and each contains these nodes:

- Fields
- Pages
- Actions and Links
- Server Scripts

Here are brief statements of each node’s purpose:

- Use the Fields node to modify standard fields and create new fields.
- Use the Pages node to create and modify page layouts.
- Use the Actions and Links node to create internal actions or links to external applications with Groovy scripts.
• Use the **Server Scripts** node to create validation rules, triggers, and object functions with Groovy scripts.

## Pages and Page Layouts in Application Composer

Use the **Pages** node to create and modify page layouts. Page layouts define what fields users will see and modify. A page layout is a design of a page which you create, and then attach conditions to. Using conditions, you can present the same Oracle Cloud Innovation Management page differently to various users. For example, a product manager can see one version of the page, while the sales representative sees another version. Page layouts are available only for simplified pages.

To create a new layout:

- Click the **Pages** node in the desired object you want to modify.
- Click the **Duplicate** icon for the type of layout you intend to define, such as the **Details Page** layout.
- Name the layout, and click **OK**.
- In the **Summary** area, click the **Edit** icon.
- In the **Available Fields** panel, select the fields that you want to display and move them to the **Selected Fields** panel.

## Working with Page Layouts for Standard Objects

Standard objects that have simplified pages are delivered with default page layouts, called standard layouts. Standard layouts are the pristine model layouts that you can’t edit. However, you can duplicate the standard layout to create a new layout. You can edit configured layouts and add display conditions to them. When a configured layout is no longer of use, you can inactivate, or deprecate, that layout.

You can:

- **Duplicate page layouts**
  
  To create a new page layout, duplicate an existing layout and then make your edits.
  
  All supported objects are delivered with a standard layout for their simplified user interface pages. Duplicate the standard layout to configure new layouts, which you can edit.
  
  The first configured layout for a page type is automatically named the default configured layout, but you can change the name.

- **Edit configured layouts**
  
  You can edit only configured layouts. Configured layouts are duplicated from an existing layout.

- **Inactivate, or deprecate, configured layouts**
  
  You can delete a page layout from a duplicated sandbox that has not been published.
  
  You cannot delete a page layout from a published sandbox, but you can inactivate configured layouts by deselecting the Active check box for a page layout on the Simplified Pages tab.
  
  You cannot inactivate the standard layouts that are automatically delivered for an object.
Working with Page Layouts for Configured Objects

Working with page layouts for configured objects is exactly the same as working with page layouts for standard objects. The only difference is that after you create a configured object, you must manually create its set of configured layouts before you can start to work with them. Configured objects do not have a set of standard layouts.

Create a set of simplified page layouts for a configured object with a click of a button. Clicking that button tells Application Composer to automatically create the following:

- Page layouts for the object’s set of user interface pages, such as the creation and details pages.
- A Search and Select dialog, which you can configure by clicking the Edit Picker link.

Standard and Configured Layouts

Standard layouts are the pristine model layouts that you can’t edit. Configured layouts are copies of standard layouts that you make, which you can edit.

Standard layouts exist to make your upgrades seamless. When you upgrade to a new release of Oracle Cloud Innovation Management, Oracle upgrades only the standard layouts for each object. Your configured layouts are not touched. This makes it easy for your users to continue working immediately after an upgrade. In the meantime, you can take your time to review the changes that happened to standard layouts as part of an upgrade, and manually incorporate those changes as and when needed.

Fields in Application Composer

Using Application Composer, you can extend Oracle Innovation Management Cloud or Quality Management applications by adding new fields to both standard or configured objects. The fields that you add to an object are configured fields. When creating a configured field, Application Composer provides a set of field types that you can choose from. For example, you can create a Check Box field, or create a Long Text field.

Use the Fields node to modify standard fields and create new fields. The following field types are available for use, and all the field types are detailed in this topic. The several rows of fields in Application Composer’s Fields node are replicated here; the field types are discussed in pairs of similar types.

<table>
<thead>
<tr>
<th>Text</th>
<th>Numbers</th>
<th>Date</th>
<th>Long Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Box</td>
<td>Percentage</td>
<td>Datetime (that is, Date And Time)</td>
<td>Record Type</td>
</tr>
<tr>
<td>Choice list (fixed)</td>
<td>Choice list (dynamic)</td>
<td>Formula</td>
<td></td>
</tr>
</tbody>
</table>
Adding a New Field to a Business Object

Here is a short procedure to add a field to an object.

1. Confirm that you’re in a sandbox session, before making any changes to Oracle Innovation Management Cloud.
2. In Application Composer, select the object that you want to make changes to, then select the object’s Fields node.
3. On the configured Fields tab, click **New**.
   
   Application Composer provides a set of field types that you can choose from when creating new fields:
   
   - **Text**
   - **Long text**
   - **Number**
   - **Percentage**
   - **Date**
   - **Datetime**
   - **Check box**
   - **Formula**
   - **Fixed choice list**
   - **Dynamic choice list**
   - **Record Type**

4. Select the type of field you want to create, and then specify the required field attributes to create the configured field.
5. After you create configured fields, you must expose those fields on the user interface pages, before your end users can see them. See Defining Pages: Explained.

When you create configured fields for objects and expose the fields on desktop pages, Application Composer automatically creates most underlying object artifacts. When configured fields are published, full Web service support is provided for those new fields, as well. Application Composer also makes it easy to enable your object model extensions for importing and exporting.

The following properties are common across multiple field types:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Appearance</td>
</tr>
<tr>
<td>Help Text</td>
<td>Appearance</td>
</tr>
<tr>
<td>Display Width</td>
<td>Appearance</td>
</tr>
<tr>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
<td>Name</td>
</tr>
<tr>
<td>Required</td>
<td>Constraints</td>
</tr>
</tbody>
</table>
Text and Long Text field types

One field type is for **Text** fields, where users can enter a combination of letters, numbers, or symbols.

Create a text field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the Text field type.

The following properties are particular to text fields:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display Type</strong></td>
<td>Appearance</td>
</tr>
<tr>
<td>The way you want this text field to render in the application:</td>
<td></td>
</tr>
<tr>
<td>• As a simple text box.</td>
<td></td>
</tr>
<tr>
<td>• Allowing multiple lines where text can wrap or where the user can enter carriage returns.</td>
<td></td>
</tr>
<tr>
<td><strong>Maximum Length</strong></td>
<td>Constraints</td>
</tr>
<tr>
<td>The maximum number of characters that a user can enter in the field. You can set a maximum length of 1500 characters. If the field is a multiline field, then carriage returns are permitted and count as part of the total.</td>
<td></td>
</tr>
<tr>
<td><strong>Minimum Length</strong></td>
<td>Constraints</td>
</tr>
<tr>
<td>The minimum number of characters that a user can enter into the field.</td>
<td></td>
</tr>
</tbody>
</table>

Another field type similar to Text is the **Long Text** field type, where users can enter a combination of letters, numbers, or symbols. This field type supports 32,000 characters.
Create a long text field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the long text field type.

The following properties are particular to Long Text fields:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display Type</strong></td>
<td>Appearance</td>
</tr>
</tbody>
</table>

Indicate how you want this text field to render in the application:

- As a simple text box.
- Allowing multiple lines where text can wrap, or where the user can enter carriage returns.

**Number and Percentage field types**

One field type is the **Number** field type, where users can enter a number.

Create a number field by specifying values for the common set of field properties, such as Display Label and Field Name. You also set properties for this field that are specific to the Number field type.

The following properties are particular to Number fields:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decimal Places</strong></td>
<td>Constraints</td>
</tr>
</tbody>
</table>

The number of digits that can be entered and displayed after the decimal point. If at run time, a user enters more digits after the decimal point, then Application Composer rounds up to derive the field's value.

For example, if you enter 2 for the number of decimal places, then at run time, an entry of 4.986 is displayed as 4.99.

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum Length</strong></td>
<td>Constraints</td>
</tr>
</tbody>
</table>

The number of digits a user can enter in the field. This number should be greater than or equal to 1 and less than or equal to 38.

During field creation, consider how this property interacts with these other field properties:

- **Display Width**
  If you set a maximum length that is longer than the display width, then users must scroll inside the field at run time to see the number in this field.

- **Decimal Places**
  **Maximum Length - Decimal Places** = the number of digits that can appear before the decimal point.
  Do not set a maximum length that is shorter than the number of decimal places.
Another field type somewhat like Number is the **Percentage** field type, where users can enter a percentage. Application Composer automatically adds the percent sign to the number.

Create a percentage field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the percentage field type.

The following properties are particular to percentage fields:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decimal Places</strong></td>
<td>Constraints</td>
</tr>
<tr>
<td>The number of digits that can be entered and displayed after the decimal point. If at run time, a user enters more digits after the decimal point, then Application Composer rounds up to derive the field’s value.</td>
<td>For example, if you enter 2 for the number of decimal places, then at run time, an entry of 4.986 is displayed as 4.99.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum Length</strong></td>
<td>Constraints</td>
</tr>
<tr>
<td>The maximum digits a user can enter in the field.</td>
<td>During field creation, consider how this property interacts with these other field properties:</td>
</tr>
</tbody>
</table>

- **Display Width**
  If you set a maximum length that is longer than the display width, then users must scroll inside the field at run time to see the amount in this field.

- **Decimal Places**
  Maximum Length - Decimal Places = the number of digits that can appear before the decimal point.
  Do not set a maximum length that is shorter than the number of decimal places.

**Date and Datetime field types**

One field type is a **Date** field, where users can enter a date or select one from a calendar. This type of field has no time component. Create a date field by specifying values for the common set of field properties, such as display label and field name.

Another field type similar to Date is the **DateTime** field type, where users can enter a date OR select one from a calendar, AND enter a time of day. Create a datetime field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the datetime field type.

The following properties are common across Date and Datetime field types:
Check Box and Formula field types

One field type is the Check Box field type, which users can select it to indicate a record’s true or false attribute.

Create a check box field by specifying values for the common set of field properties, such as display label and field name.

The following properties are common across multiple field types:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Appearance</td>
</tr>
<tr>
<td>Help Text</td>
<td>Appearance</td>
</tr>
<tr>
<td>Display Width</td>
<td>Appearance</td>
</tr>
<tr>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
<td>Name</td>
</tr>
<tr>
<td>Required</td>
<td>Constraints</td>
</tr>
<tr>
<td>Updatable</td>
<td>Constraints</td>
</tr>
<tr>
<td>Fixed Value</td>
<td>Default Value</td>
</tr>
<tr>
<td>Expression</td>
<td>Default Value</td>
</tr>
</tbody>
</table>
Another field type is the **Formula** field type, which is calculated using the Groovy-based expression included in the field’s definition.

Create a formula field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the formula field type.

The following properties are particular to Formula fields:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formula Type</td>
<td>Field Value Type</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Formula Type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specifying the field’s data type, such as text, number, or date. You can specify the type only during field creation.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Display Type</td>
<td>Appearance</td>
</tr>
<tr>
<td></td>
<td>If the formula type is Text, then indicate if you want this formula field to render in the run time application as a simple text box, or if the field should allow multiple lines where text can wrap.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Depends On</td>
<td>Constraints</td>
</tr>
</tbody>
</table>

Additional specifications for this field type include the following details:

- Data type is set by the **Formula Type** property.
- The formula field type is not supported by configured subject areas. You cannot add formula fields to a configured report.
- You cannot search on a formula field.
- A formula field is a computed attribute, and exists only at run time. This is a transient type of attribute that does not persist in the database as a table column. Hence, no maximum number of formula fields exists for an object.
- A formula field’s Groovy script is evaluated every time the field’s value is requested by any layer. You should not use a formula field to set other fields’ values because, due to the order of rendering, the order in which the fields are processed is not guaranteed.

**Choice Lists - Fixed and Dynamic - field types**

One field type is a **Fixed Choice List**, a field that contains a list of static values. At run time, users can select one or more values from this field, depending on the field’s definition.

Create a fixed choice list by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the fixed choice list field type.

The following properties are common across multiple field types:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Appearance</td>
</tr>
</tbody>
</table>
Field Property | Field Property Region
---|---
Help Text | Appearance
Display Width | Appearance

The size of the field depends on the longest value of the strings in the choice list.

Name | Name
Description | Name
Required | Constraints
Updatable | Constraints
Searchable | Constraints

**Fixed Value**

You cannot set a default value for any fixed choice list that is constrained by another fixed choice list.

For example, if the field includes three lookup codes with (Code,Label) of (S,Small),(M,Medium), (L,Large), and (XL,Extra Large), then to preselect the Small and Extra Large selections by default, set the default value to the literal string (without quotes): \textbf{S,XL}.

If the choice list allows multiple values, you can still write an expression to preselect multiple values by default.

The data for the multi-select pick list is stored in comma-separated format; in the previous example, "S,XL" will be stored in the database.

The following properties are particular to Fixed Choice Lists:

Field Property | Field Property Region
---|---
Display Type | Appearance

Indicate if users can select a single value or multiple values from the choice list at run time. You can only select the display type during field creation.

\textbf{Note}: If you create a multiple-select fixed choice list, then do not use commas in the lookup codes that populate this field.

Lookup Type | List of Values
Constrain List by Parent Field Value Selection | You cannot create a Lookup Type with a name ending in "LOOKUPTYPE". If you do, you won’t be able to see this extension in BI Answers and reporting.

**Selecting the List of Values for the Fixed Choice List**
The values in a fixed choice list are populated from Lookup types. Select the lookup type with values you want to display in this choice list. You can only select the lookup type during field creation. A fixed choice list can have a maximum of 1,000 values.

You can also create a new lookup type and add new values to it. Enter a lookup type and select the Edit icon to modify the existing values.

The set of Lookup types available for selection is constrained to those lookup types related to this fixed choice list’s object (by product code), as well as all configured lookup types that have been created for your Cloud implementation.

**Dynamic Choice List**

Another field type similar to Fixed Choice List is the **Dynamic Choice List** field type. A dynamic choice list is a field that contains a list of values that are populated from the actual data of another object. You can add a dynamic choice list to a configured or standard object.

For example, you might want to expose on a user interface page a dynamic choice list which lets users specify the account that they are logging a help request against. In this example, the **Account Name** choice list is a field that you are adding to the help request object, but the list of values is populated by actual names from the account object.

When creating dynamic choice lists, note the following:

- Review the common set of field properties, as well as the dynamic choice list-specific properties, that you must specify.
- Review the options available in the List Data Source, Additional List Display Values, and Additional List Search Fields regions.
- Understand how a dynamic choice list results in the implicit creation of a relationship.

> **Note:** You must create a Select and Search dialog box (picker) for a configured object, if you also create a dynamic choice list that is based on the same configured object.

**Record Type field type**

One field type that you can add is a **Record Type** field. A record type field contains a list of static values that are populated from Lookup types. This type of field is useful, because you can associate each choice list value with a role or a page layout. This makes a record type field more powerful than a fixed choice list field or a dynamic choice list field.

Create a record type field, so that you can associate each choice list value with a role or a page layout.

You can:

- Associate each choice list value with a role.
  
    a. You do this while you are creating the field.
    
    b. At run time, when an end user views the list of values in the field, the available choices are constrained according to the user’s role.

    configured roles, which are copies of the predefined roles that Oracle provides for all customers, are displayed by default. However, you can optionally choose to display predefined roles, as well.

- Associate each choice list value with a page layout.
  
    a. You do this by adding the field to a simplified page layout, after you have created the field.
    
    b. You must then assign a choice list value to the page layout.
c. At run time, when an end user selects a value from the field, the page display changes to match the simplified page layout that you associated with the choice list value.

Note: You can create only one record type field per object, and only once the object has a work area. If the object’s work area hasn’t been created yet, then you must create the work area first, before creating the record type field.

Create a record type field by specifying values for the common set of field properties, such as display label and field name. You also set properties for this field that are specific to the record type field type.

The following properties are common across multiple field types:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Appearance</td>
</tr>
<tr>
<td>Help Text</td>
<td>Appearance</td>
</tr>
<tr>
<td>Display Width</td>
<td>Appearance</td>
</tr>
</tbody>
</table>

Note: The size of the field depends on the longest value of the strings in the choice list.

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
<td>Name</td>
</tr>
<tr>
<td>Required</td>
<td>Constraints</td>
</tr>
<tr>
<td>Updatable</td>
<td>Constraints</td>
</tr>
<tr>
<td>Searchable</td>
<td>Constraints</td>
</tr>
<tr>
<td>Fixed Value</td>
<td>Default Value</td>
</tr>
</tbody>
</table>

The following properties are particular to Record Type fields:

<table>
<thead>
<tr>
<th>Field Property</th>
<th>Field Property Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup Type</td>
<td>List of Values</td>
</tr>
<tr>
<td></td>
<td>Selecting the lookup type is possible only during field creation.</td>
</tr>
<tr>
<td>Available Record Types</td>
<td>Roles</td>
</tr>
</tbody>
</table>
Field Property | Field Property Region
--- | ---
Indicate the choice list values that each role has access to. | For example, perhaps the sales representative can see only selected choice list values, but the sales manager can see all the choice list values.

**Default Record Type**

<table>
<thead>
<tr>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicate the choice list value that each role will see by default at run time.</td>
</tr>
</tbody>
</table>

The values in a record type field are populated from Lookup types. Select the lookup type whose values you want to display in this choice list.

You can also select a lookup type and select the Edit icon to modify the existing values.

> Note: The lookup types available for selection are limited to:

1. Standard lookup types that are related to this record type field’s object (by product code).
2. All configured lookup types that have been created for your Oracle Cloud implementation.

Or, create a new lookup type and add new values to it.

### Actions and Links in Application Composer

With Application Composer, you can add actions, such as scripts, and buttons to detail pages, list pages, and others. You can also create special fields, rendered as links, that are displayed near specific fields throughout the Oracle Innovation Management Cloud or Quality Management application. Use the **Actions and Links** node to create internal actions or links to external applications with Groovy scripts.

#### Actions and Links

An action can be based on a script (a Groovy method that is defined on the object) or a URL. After you create an action, it can be exposed as a button or an option on an **Actions** menu. After you create a link, it can be selected as a field for display at run time.

A button can perform an action or navigate the user to another page in the run time application, or to another Web site. For example, you might want to provide a static link from an overview page to a corporate Web site. Or, you might want to include a button on a summary table, which users can click at run time to create a new type of record from a selected row, such as escalating an existing "trouble ticket" to a more severe "case" that can be managed separately.
Note: Do not create new buttons to populate the mandatory or required fields on the user interface. End users must enter the values in the mandatory fields manually. When configuring the work area for a standard or configured object, you can follow these strategies:

- Add new Actions or links to a page-level or task-level Actions menu, or as a toolbar button.
- Manage the Actions menu by hiding or showing menu items, rearranging the action groupings or display sequence, and managing the toolbar by hiding or showing icons and buttons.
- Configure the Actions menu and buttons in the Create and Edit subtabs and tree nodes.

When displaying actions as buttons, be sure to test your page at run time (in all supported languages) to confirm that the presentation of buttons is as expected. Button display may have an unexpected result, which can be due to:

- the available space on the page at run time,
- the number of buttons on the page, or
- button width (which depends on label length).

If you add more buttons than the toolbar has space, then the buttons are stacked at run time and made available using a list button.

Adding Actions or Links: Overview

You add actions or links in two steps:

1. Define an action or link for an object.
2. Use Application Composer’s work area configuration pages to add that action or link to an Overview page or Details page.

Defining Actions or Links

To define an action or link for an object:

1. On the main Overview page in Application Composer, select a standard or configured object in the object tree.
2. Select the Actions and Links node.

To create a new script or URL:

1. In the Create Action or Link page, enter a descriptive name in the Display Label field.
2. For Type, select Action and, for Source, select Script or URL.
3. In the Script region click the New icon.

Exposing Actions or Links on Pages

After you save actions or links, you can expose them on user interface pages by configuring Application Composer options available in the Edit Summary Table page in the Pages node of an object.

Note: If you define a configured action and expose it on a list, ensure that you include a check for active record row, and that the user interface supports users selecting any record as the active row before invoking the configured action.

After you define an action, you can then expose it as a button or an Actions menu option in a variety of locations:

- On simplified pages
- Summary table on the overview page
- Default summary on the details page
• Summary table on a details page's subtab
• Summary table on a tree node page for a child object
• Revenue table on the details page for the opportunity object

After you define a link for an object, you can add that link to a variety of locations in that object's work area. You can add a link wherever you can add a field. Possible locations include, but are not limited to:

• As a column in the summary table on the overview page
• Default summary on the details page
• As a column in the summary table on a details page's subtab
• In the detail form, from the summary table on a details page's subtab
• As a column in the summary table on a tree node page for a child object
• As a column in the revenue table on the details page for the opportunity object

Server Scripts

Groovy is a standard, dynamic scripting language for the Java platform for which Application Composer provides deep support. This topic provides an overview of where you can use groovy in your application and gives some samples of one or more lines of Groovy code. You can also find information on Supported Classes and Methods for Use in Groovy Scripts and some examples in the Related Topics section.

Use the **Server Scripts** node to create validation rules, triggers, and object functions with Groovy scripts.

Groovy Scripting Terminology

Throughout the document the term script is used to describe one or more lines of Groovy code that the Oracle ADF framework executes at run time. Often a very-short script is all that is required.

For example, to validate that a standard or custom field’s value does not exceed 40, you might use a one-line script like:

```
return IdeaVoteCount < 40
```

In fact, this one-liner can be conveniently shortened by dropping the return keyword since the return keyword is always implied on the last line of a script:

```
IdeaVoteCount < 40
```

For slightly more complicated logic, your script might require some conditional handling. For example, suppose the maximum Idea Vote Count is 40 for voters of age 30 or less, but 60 for voters whose age is higher. Your script would grow a little to look like this:

```
if (VoterAge <= 30) {
    return IdeaVoteCount < 40
} else {
    return IdeaVoteCount < 60
}
```

Scripts that you write for other purposes like complex validation rules or reusable functions may span multiple pages, depending on your needs.

When a context requiring a Groovy script will typically use a short (often, one-line) script, we emphasize that fact by calling it an expression. Technically the terms script and expression are interchangeable. Anywhere you can provide a one-line
expression is also a valid context for providing a multi-line script if the need arises. Whether you provide a short expression or a multi-line script, the syntax and features at your disposal are the same. You need only pay attention that your code returns a value of the appropriate type for the context in which you use it.

The Groovy Scripting: Examples topic includes all the return types. This topic highlights the expected return type for each script example.

Using Groovy Scripts in Your Application

There are a number of different contexts where you will use Groovy scripts as you modify existing objects or create new configured ones.

You will write shorter scripts to provide an expression to:

- Calculate a configured formula field's value
- Calculate a configured field's default value
- Make a configured field conditionally updatable, or
- Make a configured field conditionally required
- Define the condition for executing an object workflow

You will generally write somewhat longer scripts to define:

- A field-level validation rule
- An object-level validation rule
- A trigger to complement default processing
- Utility code in a global function, or
- Reusable behavior in an object function

If you anticipate calling the same code from multiple different contexts, any of your scripts can call the reusable code you write in either global functions or object functions. As their name implies, global functions can be called from scripts in any object or from other global functions. Object functions can be called by any scripts in the same object, or even triggered by a button in the user interface.

Configure User Interface with Application Composer

This topic presents procedures of modifying user interface in several interesting ways. Working continuously through procedures help you manage these functions with ease - be it creating or removing interface fields, tabs, and layout regions and pages. You make it a point to always start in a sandbox environment, move to the Application Composer and your live applications, and learn when it is best to start over on a project and to resolve means toward objectives that didn't quite work. Over time you will rapidly develop successful modifications that help give ideas for further projects.

While projects in this and other Application Composer topics are complete from start to finish, it is always best to remember this important sequence whether it is explicitly stated or not:

1. Confirm that you're in a sandbox session, before making any changes to business objects in Oracle Innovation Management Cloud or Quality Management.
2. When prepared with an objective in live application, navigate to the Application Composer.
3. From the Application list, select ERP and SCM Cloud.
4. Click the **Innovation** check box or **Quality** check box - or both if you have both suites or applications. If you do not have an application available, you will not see its business objects in Application Composer.

5. Select the object that you want to make changes to, and review what specific configuration task you intend to accomplish.

### How You Hide Tabs and Fields

In this procedure, you can use Application Composer to hide tabs and fields on a page layout.

1. Navigate to Application Composer and set the conditions to work with a particular business object, in this case **Proposals**.
2. Expand the **Proposals** node, and click the **Pages** link.
3. Click the **Duplicate** icon on the **Details Page Layouts** section.
4. Enter a name for your new layout. The name should remind you what its purpose or highlight is. For example, this layout might be called **Hide Tabs_01** or **Hide Tabs_Proposals_01**.
5. From the **Source Layout**, select **Department Locations**.

Now click **Save and Edit**.

6. Hide one or two tabs.
   - Click the **Hide, Show, Reorder Subtabs**.
   - Move **Cost, Revenue, and Resources** to **Available Subtabs**.
   - Click **OK**.

7. Now, remove one or two fields.
   - Click the **Edit** icon in the Summary section. Move Departments and Locations to **Available Fields**.
   - Click **Save and Close**.
   - Click **Done**.

8. Add a User Account Condition.
   - On the **Hide Tabs** layout row, click the Advanced Expression icon.
   - Enter the following script. The example user name **AC_admin** can be any User Name that is valid in your Oracle SCM Cloud suite of applications.
     ```python
     def user
     user = adf.context.getSecurityContext()?.getUserName()
     user == AC_admin
     ```
   - Click **OK**.

9. Test the Layout.
   - Navigate to the **Concept Design** work area.
   - Expand the **Search** panel.
   - Select **Proposals** in the **Search For** field.
   - Click the **Search** icon.
   - Click a proposal link.
   - Verify that the tabs and fields are no longer visible.
How You Add a Text Field

In this procedure, you create a Text field and add it to a page layout.

1. In a sandbox environment or not, navigate to Application Composer and set the conditions to work with a particular business object, in this case Ideas. Expand the Ideas node, and click the Fields link.

2. Create a text field called Input.
   - Click the Fields link. On the Custom tab, click the Create icon.
   - Select the Text radio button and click OK.
   - Enter a name in the Display Label field, for example “In-put_01”.
   - Enter some input in the Help Text field. Click the Save and Close button.

3. Create a new Details Page layout entitled Default layout.
   - Click the Pages link in the panel.
   - Click the Duplicate icon on the Details Page Layouts section. Change the name of the layout to Default Layout_01.
   - Click the Save and Edit button on the Duplicate Layout pop-up.
   - Click the Default Layout_01 link. Click the Edit icon in the Summary section of the General Information tab.
   - Select the In-put_01 field and move it to Selected Fields.
   - Click Save and Close, and click Done.

4. Test the new fields in an idea.
   - Navigate to Ideas and click the Manage Ideas tab.
   - Click the link of any Idea. Enter some text in the In-put_01 field.
   - Notice that clicking on the field displays the help string.

How You Use a Formula Field

In this procedure, you create two Number fields and a Formula field. You will enter a formula to multiply the two number values and place all of these fields on a new page layout.

1. In a sandbox environment or not, navigate to Application Composer and set the conditions to work with Ideas. Expand the Ideas node, and click the Fields link.

2. Add a Number field called Length_01.
   - Click the Create icon.
   - Select Number as the Field type and click OK.
   - Enter the following as the Display Label: Length_01.
   - Enter 10 as the Display Width.
   - Click Save and Close.

3. Add a Number field called Width_01.
   - Click the Create icon.
   - Select Number as the Field type and click OK.
   - Enter the following as the Display Label: Width_01.
4. Add a Formula field called Square Feet_01.
   - Click the Create icon.
   - Select Formula as the field type and click OK.
   - Select Number as the Formula Type.
   - Enter the following as the Display Label: Square Feet_01
   - Enter 10 as the Display Width.
   - Select Length_01 and Width_01 in the Depends On field.
   - Click Next.
   - Enter the following function: \( \text{nvl(LengthXX_c,0.00)} \times \text{nvl(WidthXX_c,0.00)} \)
   - Click Submit.

5. Create a new Details Page layout entitled Default Layout.
   - Click the Pages link in the panel.
   - Click the Duplicate icon on the Details Page Layouts section.
   - Change the name of the layout to Default_01.
   - Click the Save and Edit button on the Duplicate Layout pop-up.

6. Modify the new layout to include the three new fields.
   - Click the Default_01 link.
   - Click the Edit icon in the Summary section of the General Information tab.
   - Select the Length_01 field and move it to Selected Fields.
   - Select the Width_01 field and move it to Selected Fields.
   - Select the Square Feet_01 field and move it to Selected Fields.

7. Click Save and Close, and click Done.

8. Test the new fields in an idea.
   - Navigate to Ideas and click the Manage Ideas tab.
   - Click the link of any Idea in the column.
   - Enter a number in the Length_01 field.
   - Enter a number in the Width_01 field.
   - Notice that the Square Feet_01 field was automatically calculated.

Create a Requirements Specification Using Groovy Script

Video

Watch: This video demonstrates how to create objects using groovy script. The content of this video is also covered in text topics.
Procedure

This procedure helps you create Innovation Management objects - ideas, portfolios and requirements specifications - using groovy script. Let’s look at how you can create a requirements specification using a groovy script.

1. Navigate to the Application Composer.

   Note: Ensure that you are in a sandbox.

2. Select the ERP and SCM Cloud option from the Application list.
3. Select the Innovation check box in the Object tags.
4. Expand Standard Objects > Ideas > Server Scripts to create an object function script.
5. To create an object function script:
   b. On the Create Object Function page that opens, enter a name in the Function Name field. For example: createRequirementsSpecification
   c. In the Edit Script panel, enter groovy script details:
      
      ```groovy
      def specView = newView('RequirementSpecification')
      def specObj = specView.createRow()
      specObj.setAttribute('Type', "TEST_CASE")
      specObj.setAttribute('Name', "Test_F19")
      specView.insertRow (specObj)
      ```
   d. Click Save and Close.

Let’s create an action link, since the created object functions work only if attached to an action.

6. To create action links:
   a. Click Action and Links.
   b. On the Idea: Create Action or Link page, click the Create button. The display label is what you see on the button that you click. Enter a value. For example: Create Requirements Specification. Internal names are automatically filled.
   c. Select the Method Name that you created earlier in the Server Script node.
   d. Click Save.

Since the action works only if it’s added to a page layout, let’s add the action to a page.

7. To add the action to a page:
   a. Click Pages.
   b. On the Idea: Pages page, go to Details Page Layouts panel and click the Duplicate button to create a duplicate layout.
   c. Enter a name in the New Layout Name field For example: Default custom layout.
   d. Click Save and Edit.
   e. On the Details Layout: Default custom layout page, click Edit icon next to the Actions menu. You can see that the action name is available as a button and as an action.
   f. Move the action to the Selected Buttons or Selected Actions panel.
   g. Click Save and Close.
   h. Click Done.
8. To check if the settings that you set work:
   a. Navigate to the Ideas work area.
   b. Click the Post Idea button.
   c. In the Post idea dialog box, name the idea and click Save and Close.
   d. Navigate to the Concept Design work area.
   e. In the Tasks panel select Manage Requirements Specifications and search for Test_F19. Notice that the search returns your requirements specification.

You can also create Proposals and Concepts using groovy script.

FAQs on Configure the User Interface with Application Composer

What job role must I have to create my own objects in Application Composer?

Users with any one of the three following job roles can create their own objects and use all other Application Composer functions:

- Customer Relationship Management Application Administrator.
- Application Implementation Consultant.
- Master Data Management Application Administrator.

Oracle recommends provisioning the user with the Customer Relationship Management Application Administrator job role (for performing the configurations) and the Custom Objects Administration job role and Sales Administrator job role (for testing the configurations).

What's the difference between fixed choice lists and dynamic choice lists?

A fixed choice list and a dynamic choice list are similar in that the ultimate goal of both types of choice lists is to generate a field with a list of values at run time. However, the list of values for a fixed choice list is derived from an FND lookup type. The list of values for a dynamic choice list is derived from an existing object’s actual data.
What Application Composer tasks are available only within a sandbox?

Most Application Composer tasks require you to be in a sandbox. For example, these menu items are available to you only if you’re in an active sandbox session.

- Objects
  - Custom Objects
  - Standard Objects
- Common Setup
  - Relationships
  - Role Security
  - Object Workflows
  - Global Functions
  - Run Time Messages
  - Mobile Application Setup
  - Outlook Setup
  - Personalization
  - Web Services
  - Metadata Manager

These menu items are the exceptions. They’re available only in a sandbox-free session.

- Custom Subject Areas
- E-Mail Templates
- Import and Export
- Business Processes

When do I publish a sandbox?

You can publish a sandbox after you have tested and verified that the application changes done in that sandbox are ready to be moved to the mainline metadata.

You must test the following configurations outside a sandbox:

- Import/Export
- Web services
- Custom subject area creation
- Object workflow
- E-mail templates
How frequently can I publish a sandbox?

Integration sandboxes are typically published once a week. Publishing integration sandboxes less frequently than once a week isn’t recommended.

When you publish an integration sandbox, all private sandboxes are invalid because the label in the mainline metadata application has changed. If you made changes to private sandboxes that you want to retain, then document those changes and then delete all the private sandboxes.

Can I delete a sandbox?

Yes. You can delete sandboxes. However, you can delete only those that aren’t published.

Before you delete a sandbox, you must first confirm that the sandbox isn’t active.

⚠️ Caution: Deletion of partial content of a sandbox is risky. It’s recommended that you don’t use this option.

After you have tested your application changes, you must move those changes to the integration sandbox. Publish your integration sandbox and then delete all the test-only sandboxes. You can then create and work in new sandboxes, including a new integration sandbox.

What’s the difference between Page Composer and Application Composer?

Page Composer is a web-based tool you can use to modify user interface (UI) pages and components for all products designated for use with Page Composer. Page Composer uses two different modes of Design View. The first mode, Design View: Standard mode, is selected by default in all the pages when opening a page with Page Composer with the Design button selected. The second mode, Design View: Direct Selection mode, is activated when you click the Select tab for the UI page you want to modify. Direct Selection mode is available when you modify pages, but not when you personalize a dashboard page. With the Design View: Direct Selection mode, you can select and edit UI elements such as form fields and table columns. In Direct Selection mode, the UI components that you can select become apparent when you move your cursor over them. The UI components that you can select are highlighted and can be edited.

The following table describes how you can use each mode of Page Composer to modify dashboard pages and other select pages (such as the Partner Public Profile page, Partner Landing page, Partner Registration, Customer Snapshot, and Customer Overview - Analysis tab), and modify transactional pages (all other non-dashboard pages).

<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Design View - Standard mode</th>
<th>Design View - Direct Selection mode</th>
<th>Page Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add content (Business Intelligence reports, portlets such as Calendar)</td>
<td>Yes</td>
<td>No</td>
<td>Dashboard and other select pages</td>
</tr>
<tr>
<td>Delete region</td>
<td>Yes</td>
<td>No</td>
<td>Dashboard and other select pages</td>
</tr>
</tbody>
</table>
### Application Composer Features and Use Cases

<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Design View - Standard mode</th>
<th>Design View - Direct Selection mode</th>
<th>Page Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move region</td>
<td>Yes</td>
<td>No</td>
<td>Dashboard and other select pages</td>
</tr>
<tr>
<td>Change page layout (for example, change a two column layout to three column layout)</td>
<td>Yes</td>
<td>No</td>
<td>Dashboard and other select pages</td>
</tr>
<tr>
<td>Default region state (open or close)</td>
<td>Yes</td>
<td>No</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Manage save queries (create and edit)</td>
<td>Yes</td>
<td>No</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Hide or show field</td>
<td>No</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Change field label</td>
<td>No</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Make field required or not</td>
<td>No</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Make field read-only or updatable</td>
<td>No</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Reorder fields in a Form</td>
<td>No</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Reorder table columns</td>
<td>Yes</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Hide or show table columns</td>
<td>Yes</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Set table column width with the mouse</td>
<td>Yes</td>
<td>No</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Set table column width and minimum width in percent or pixels</td>
<td>No</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
<tr>
<td>Sort column or not</td>
<td>No</td>
<td>Yes</td>
<td>Transactional pages (all non-dashboard pages)</td>
</tr>
</tbody>
</table>

Application Composer also lets you make UI changes at run time. However, the types of UI changes that you can make using Application Composer are quite different. Specifically, your primary focus when using Application Composer is to make actual object model changes. For example, you can create a new business object and related fields, and then create new application pages where that object and its fields are exposed to users.
The following table describes some of the primary differences between Page Composer and Application Composer. For example, using Application Composer, you cannot access the Resource Catalog to add new content to a page.

<table>
<thead>
<tr>
<th>Task</th>
<th>Available in Page Composer (site, job role, external or internal level)?</th>
<th>Available in Application Composer (site level only)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make object model extensions and expose your changes by creating or modifying work area pages</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Reorder subtabs</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Modify dashboard pages</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Add content from the Resource Catalog</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Simple field changes (show, hide, make read only, make required)</td>
<td>Yes (WYSIWYG - what you see is what you get)</td>
<td>Yes (non-WYSIWYG)</td>
</tr>
<tr>
<td>View results of changes immediately</td>
<td>Yes, in the Page Composer design interface</td>
<td>Yes, in the application that you are making changes to</td>
</tr>
</tbody>
</table>

**Related Topics**

- About Application Composer
4 Configure Innovation Management for Integration with External Systems

Integration Tasks

The tasks Register Agile PLM, and Manage Target System aren’t module-specific. Complete these optional tasks, listed in the following table, as required to integrate external PLM systems with Oracle Innovation Management.

*Note:* You must first complete the common application setup and configuration tasks for Product Management in the Setup and Maintenance workspace.

<table>
<thead>
<tr>
<th>Task List</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register Agile PLM</td>
<td>This task is a prerequisite for the Manage Target System task, and required to connect Oracle Agile Product Lifecycle Management (Agile PLM) to Oracle Innovation Management.</td>
</tr>
<tr>
<td>Manage Target Systems</td>
<td>Use this task to configure connections between Oracle Innovation Management and external Product Lifecycle Management (PLM) systems, or Oracle Product Development.</td>
</tr>
</tbody>
</table>

Register Agile PLM

This task is a prerequisite for the Manage Target System task, to integrate Oracle Agile PLM with Oracle Innovation Management.

*Note:* You must first complete the common application setup and configuration tasks for Product Management in the Setup and Maintenance workspace.

The typical Agile PLM endpoint is {protocol}://{host}:{port}/CoreService/services/{service}?wsl

The values for protocol, host, port, and context root (CoreService) need to be entered into the corresponding filed for registering the target endpoint in your Cloud application using the Setup and Maintenance workspace.

Use the Register Agile PLM task, as described in the following table, to configure server details of the target system (Agile PLM) intended for use.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Server Protocol</td>
<td>Select from the menu options (http or https)</td>
</tr>
<tr>
<td>*External Server Host</td>
<td>Enter the Agile PLM system name.</td>
</tr>
<tr>
<td></td>
<td>Example: &lt;plmserver&gt;. oracle.com</td>
</tr>
</tbody>
</table>
The following table details the Associated Modules and their Context Root Values.

<table>
<thead>
<tr>
<th>Module</th>
<th>Context Root Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgileA9WebClient</td>
<td>Agile</td>
</tr>
<tr>
<td>AgileA9CoreService</td>
<td>Enter the SAML web service reference value here that matches the value defined in the file application.xml.</td>
</tr>
</tbody>
</table>

Related Topics

- Configure Message Protection for Agile PLM

Manage Target System

Use the Manage Target System task to configure data connections, based on web services, between Oracle Innovation Management and target PLM systems.

- Use the Manage Connections tab to define template-based connectors that enable access between Oracle Innovation Management and target PLM systems or Oracle Product Development.
- Use the Manage Mapping to External System tab to configure the display and usage of target PLM or Oracle Product Development Cloud entities and their attributes within Oracle Innovation Management.
- Use the Manage Mapping to Innovation Management tab to map Oracle Innovation Management entities and attributes to target PLM or Oracle Product Development Cloud entities, depending on the connector used.

Related Topics

- Integration with External Systems

Manage Connections

Use the Manage Connections task to configure multiple target PLM system connections, and activate any one at a given time.

Preconfigured connector templates are available for Agile PLM, and Product Development configurations. The connector templates are easily identified by their names (beginning with ORA_) as well as their descriptions in the Connector Type column.
Duplicate the connector template
Duplicate any one of the existing connector templates to create a connector.

 Note: You cannot edit a connector template directly.

To duplicate a connector template:
1. Select it and click the Duplicate icon or use the Duplicate option from the Actions menu.
2. Name the newly created connector.

 Note: The connector name cannot begin with ORA_.

Configure the Agile PLM Connector
To integrate Agile PLM with Oracle Innovation Management, start by duplicating the ORA_A9 connector template. Edit the newly created connector using the Edit option from the Actions menu.
Alternatively, select the newly created connector and click the Edit icon. The following table lists and describes the connector values to configure.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client URL</td>
<td>URL of the Oracle Agile PLM Web Client</td>
</tr>
<tr>
<td>*Object Create Batch Size</td>
<td>Minimum value is 10.</td>
</tr>
<tr>
<td></td>
<td>Additionally, the maximum value cannot exceed 1,000.</td>
</tr>
<tr>
<td></td>
<td>If this value is high, the web service payload is too large.</td>
</tr>
<tr>
<td>*Object Read Batch Size</td>
<td>Define how many objects are created in Agile PLM in one chunk.</td>
</tr>
<tr>
<td></td>
<td>The minimum value is 10.</td>
</tr>
<tr>
<td></td>
<td>The maximum value depends on the hardware configuration used.</td>
</tr>
<tr>
<td></td>
<td>If this value is high, the web service payload is too large.</td>
</tr>
<tr>
<td>*Maximum Number of Search Results</td>
<td>Define the maximum number of records to be retrieved from a query, irrespective of the number of records that match the search criteria.</td>
</tr>
<tr>
<td></td>
<td>Minimum value is 10.</td>
</tr>
<tr>
<td></td>
<td>The maximum value cannot exceed 500.</td>
</tr>
<tr>
<td></td>
<td>If this value is high, the web service payload is too large.</td>
</tr>
<tr>
<td>*Buffer Size for Attachments in MB</td>
<td>Value must be between 10 and 1,000.</td>
</tr>
<tr>
<td>*Buffer Size for Thumbnails in MB</td>
<td>Value must be between 10 and 1,000.</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ECO Usage</td>
<td>Define the engineering change order processing type to use when transferring items or item structures to PLM.</td>
</tr>
<tr>
<td></td>
<td>If you select an option other than <strong>User Selection</strong>, the end user is not given an option on the engineering change order action when converting a concept component in Oracle Innovation Management to an item in external PLM.</td>
</tr>
<tr>
<td>Overwrite Web Service URL</td>
<td>The check box is enabled by default, and the predefined web service endpoint is used to access an Agile PLM system which is not SAML-enabled.</td>
</tr>
<tr>
<td></td>
<td>Disable the <strong>Overwrite Web Service URL</strong> check box to engage the Web Service URL defined through the <strong>Register Agile PLM</strong> task.</td>
</tr>
<tr>
<td>Web Service URL</td>
<td>If you enable the <strong>Overwrite Web Service URL</strong> check box, ensure that the <strong>Web Service URL</strong> you add points to a SAML-enabled Agile PLM system.</td>
</tr>
<tr>
<td></td>
<td>Example: http://&lt;plmserver&gt;:&lt;port&gt;/CoreService/services</td>
</tr>
</tbody>
</table>

**Note:** If **Overwrite Web Service URL** is enabled, you can verify if the URL provided is valid, by opening the URL in a browser. If the URL is incorrect, the WSDL file does not open in the browser. You must then change the value in **Web Service URL**.

**Configure the Product Development Connector**

To integrate Oracle Product Development with Oracle Innovation Management, start by duplicating the **ORA_PD** connector template. Edit the newly created connector using the **Edit** option from the **Actions** menu.

Alternatively, select the newly created connector and click the **Edit** icon.

Use the tasks **Manage Item Organizations** and **Manage Item Classes** in the Product Management offering to create and manage item classes, item organizations, and item templates.

**Related Topics**

- **Item Classes**

**Map to External Systems**

Use the **Manage Mapping to External System** task to define entities, entity subtypes (subclass), and their relationships to corresponding entities in external PLM.

The configuration of entities in this task determines their availability and usage in all Oracle Innovation Management integration use cases.
Edit Base Entities

The connector template you use contains a nonnegotiable list of base entities. You can’t add an entity that is absent from the list.

⚠️ **Note:** You must duplicate a template connector and modify the copy. The template connectors aren’t modifiable.

1. Select the required connector from the **Connector Name** menu to view entities associated with it.
2. Click the **New** icon or **New** option from the **Actions** menu in the **Entity** pane to add entities from a list specific to each connector.

Edit or Remove Entity Subtypes

Some entities may contain subtypes (also called **subclass** in Oracle Agile PLM) that you can configure for additional value in an entity definition, and to appear in the **Quick View** of a PLM item in Oracle Innovation Management.

⚠️ **Note:** Removing subtypes deletes all related records like assigned attributes and mappings. Deleting subtypes from an active connector can cause errors in active sessions.

Select an entity and click the **New** icon or **New** option from the **Actions** menu in the **Entity** pane, to add subtypes which are defined in the Agile PLM application for the selected base entity.

Edit an Entity

Select an entity and click the **Edit** icon or **Edit** option from the **Actions** menu in the **Entity** pane, to modify the entity name or its **auto number** source.

⚠️ **Note:** The **auto number** source is applicable to the Agile PLM connector only. Define the value in the Oracle Agile PLM Java Client for object classes, to allow newly created objects in Agile PLM to be automatically numbered.

Define Individual Attributes of Entities

Among other options, decide the order of visibility of each attribute, and if it can be found in the Oracle Innovation Management search options.

1. Select an entity from the **Entity** pane to view its attributes in the **Attribute** pane.
2. Click the **New** icon or **New** option from the **Actions** menu in the **Attributes** pane to add attributes from a list predefined per entity.
3. Select an attribute and click the **Edit** icon or **Edit** option from the **Actions** menu in the **Attributes** pane to edit it.

⚠️ **Note:** Attributes common to all subtypes, configured at the entity level in Oracle Innovation Management, are also known as **Title Block** or **Page Two** attributes in the Agile PLM Java Client. Attributes particular to some subtypes, configured at the subentity level in Oracle Innovation Management, are also known as **Page Three** attributes in the Agile PLM Java Client.

The following table lists and describes the attributes you can configure.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Name</td>
<td>Enter the name of the field as intended for display for users. The modifiable name is the label of the attribute seen in Oracle Innovation Management, such as in Quick View or the search results table.</td>
</tr>
<tr>
<td>API Name</td>
<td>The auto-generated, unique system-wide identifier for objects in Oracle Agile PLM. API Name is a read-only value field.</td>
</tr>
<tr>
<td>Data Type</td>
<td>A predefined value indicating the type of attribute. Values include: SINGLELIST_ DISPLAYVALUE SINGLELIST_ KEY MULTILIST_ DISPLAYVALUE INTEGER, DATE, STRING, MONEY_ AMOUNT, MONEY_ CURRENCY DOUBLE, UNITOFMEASURE_ AMOUNT UNITOFMEASURE_ UNIT.</td>
</tr>
<tr>
<td>Data Type Length</td>
<td>A predefined value indicating the number of bytes that you can enter in the field.</td>
</tr>
<tr>
<td>Scale</td>
<td>Enter the number of digits required after the decimal point (in a numeric field only). This setting must be greater than or equal to zero (0).</td>
</tr>
<tr>
<td>Searchable Indicator</td>
<td>Enable or disable the attribute from being added as search criteria in the Oracle Innovation Management search options for Parts, Items, Designs and Relationships. Attributes from relation entities like part structure aren’t searchable.</td>
</tr>
<tr>
<td>Search Result Sequence</td>
<td>Indicate the column order in which you require the attribute field to appear in search results within Oracle Innovation Management. Note: An empty value or a value less than 0 implies that the attribute isn’t displayed.</td>
</tr>
<tr>
<td>Quick View Sequence</td>
<td>Indicate the row order in which you require the attribute field to appear in the Quick View box that’s displayed on hovering over a PLM item within Oracle Innovation Management. Note: An empty value or a value less than 0 implies that the attribute isn’t displayed.</td>
</tr>
<tr>
<td>Advanced Search Sequence</td>
<td>Indicate the row order in which you require the attribute field to be displayed as search criteria when the Advanced Search function is in use. This field is applicable only to the Agile PLM connector template. Note: An empty value or a value less than 0 implies that the attribute isn’t displayed.</td>
</tr>
<tr>
<td>Default Value</td>
<td>Assign a default value to be used for the attribute field in Oracle Innovation Management, if the user doesn’t provide a value.</td>
</tr>
</tbody>
</table>
Configure Innovation Management for Integration with External Systems

Value | Description
--- | ---
| Default values are relevant in the following use cases:  
- In search results, when no value is defined in Agile PLM  
- When creating objects in Agile PLM.  
For example, all attributes denoting currency have a default value of \textit{USD}, as an Oracle Innovation Management user can't specify the currency type. In Oracle Agile PLM, the currency value of a business object obtained from Oracle Innovation Management is automatically appended with the value \textit{USD}.  

| Is Hover Over | Use the check box to enable or disable the attribute from being displayed as a Quick View attribute in Oracle Innovation Management.  
| Is Hyperlink | Use the check box to enable or disable the attribute from being linked directly to its source object in external PLM.  
For thumbnail attributes, a link to open the thumbnail image is provided.  
| Is Auto Number Source | Use the check box to indicate the key attribute having an auto number source.  
This field is applicable only to the Agile PLM connector template.  
To create an object of an entity in Agile PLM, you must define an auto number source value.  

Manage Mapping to Innovation Management

Use the Manage Mapping to Innovation Management task to configure value transformations for handling data across the data formats of external PLM systems, Oracle Product Development, and the data formats of Oracle Innovation Management.

Mapping Sets

Each mapping set per connector template represents a use case that filters the data model of Oracle Innovation Management to fulfill unique scenario requirements.

\textbf{Note:} You cannot edit the name of a mapping set, or create a mapping set.

The following table describes the mapping sets associated with the connector templates.

<table>
<thead>
<tr>
<th>Mapping Set</th>
<th>Use Case</th>
</tr>
</thead>
</table>
| AUCommonReferences | Control entities and type of objects that can be linked through the Relationships table  
| CDMCopyItem | Control entities when converting an item to a concept component  
The item can belong to either an external PLM system or Oracle Product Development.  
| CDMCreateItem | Control entities when converting a concept component to an item  
|
Configure Innovation Management for Integration with External Systems

### Mapping Set | Use Case
--- | ---
CDMDefault | Determine how Designs are linked through the Designs table
CDMLinkItem | Control entities when linking them through to the concept structure
CDMProposalProjects | Link projects from Agile PLM to proposals in Oracle Innovation Management.

**Note:** This mapping set is not applicable to the Oracle Product Development connector.

VINItem | Control entities in the graphical display of concept structure.

### View and Edit Entity Mappings
Select a connector and a mapping set to view and edit the entity mappings associated with the mapping set.

**Note:** You cannot add an entity that is absent from the list.

1. To add an entity to the active mapping set, click the **New** icon or **New** option from the **Actions** menu in the **Innovation Management Entity Mappings** pane.
2. To modify the **Innovation Management Entity Name**, select an entity mapping and click the **Edit** icon or **Edit** option from the **Actions** menu in the **Innovation Management Entity Mappings** pane.

You may edit the **Entity** name in the following use cases:

- Disable linking a Project from Agile PLM into Innovation Management if this use case should not be supported.
- Change the subclass of part to be used when converting components to items in Agile PLM.
- Limit the type of objects to be linked through the relationships table or rename the object names to be shown in the list.

**Note:** Changing the name incorrectly can result in errors in the data model.

3. View, add, edit, or delete attributes of the selected entity mapping.

   a. To view entity attributes in the **Attribute** pane, select an entity mapping from the **Innovation Management Entity Mappings** pane.
   b. To add attributes from a list predefined per entity, click the **New** icon or **New** option from the **Actions** menu in the **Attributes** pane.
   c. To edit an attribute, select it and click the **Edit** icon or **Edit** option from the **Actions** menu in the **Attributes** pane.

The following table describes the attributes you can configure.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Name</td>
<td>Define the <strong>Attribute</strong> name as it is to be used in Oracle Innovation Management.</td>
</tr>
</tbody>
</table>
Select from the list of attributes provided from the data definition in Manage Mapping to External System for the selected entity. The names may differ in Agile PLM.

Example: **Part Name** is used for the Agile PLM attribute **Part Number**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Innovation Management Attribute Name</td>
<td>The name of the attribute as known in Oracle Innovation Management.</td>
</tr>
<tr>
<td>Innovation Management Data Type</td>
<td>Value indicating the type of Oracle Innovation Management object attribute: DATE, DOUBLE, INTEGER, LONG, BOOLEAN, DECIMAL, STRING, JBO_NUMBER, JBO_DATE.</td>
</tr>
<tr>
<td>Innovation Management Type Length</td>
<td>Value indicating the number of numeric places or characters that can be entered in the field.</td>
</tr>
<tr>
<td>Innovation Management Type Scale</td>
<td>Enter the number of digits required after the decimal point (in a numeric field only). This setting must be greater than or equal to zero (0).</td>
</tr>
<tr>
<td>From Converter</td>
<td>Select from a list of values detailed in the section Handling Conversions Between Data Types.</td>
</tr>
<tr>
<td>To Converter</td>
<td>Select from a list of values detailed in the section Handling Conversions Between Data Types.</td>
</tr>
</tbody>
</table>

### Handling Conversions Between Data Types

Use the **From Converter** and **To Converter** options for handling conversions between data types in Oracle Innovation Management and an external PLM system.

The following table lists and describes the conversion options.

<table>
<thead>
<tr>
<th>From and To Converter Values</th>
<th>Converter Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle.apps.scm.productCollaboration.auIntegration.configuration.uiModel.AUConverterFactory $PositiveIntegerConverter.class</td>
<td>Converts negative integer values to 0; positive values remain intact.</td>
</tr>
<tr>
<td>oracle.apps.scm.productCollaboration.auIntegration.configuration.uiModel.AUConverterFactory $TolerantStringIntegerConverter.class</td>
<td>Converts strings which are numeric into positive integers (see PositiveInteger converter) and nonnumeric strings to 0 (as integer).</td>
</tr>
<tr>
<td>oracle.apps.scm.productCollaboration.auIntegration.configuration.uiModel.AUConverterFactory $TolerantStringDoubleConverter.class</td>
<td>Converts strings which are numeric into positive double values (see PositiveInteger converter) and nonnumeric strings to 0 (as double).</td>
</tr>
<tr>
<td>oracle.apps.scm.productCollaboration.auIntegration.configuration.uiModel.AUConverterFactory</td>
<td>Converts strings which are numeric into positive decimal values (see PositiveInteger converter) and nonnumeric strings to 0 (as decimal).</td>
</tr>
</tbody>
</table>
How You Enable Item Class Mapping in Product Development Connector

The PD connector supports flexible mapping of IM concept and component classes to PD item classes. In concept structures, you can also map specific user-defined attributes to item attributes in item structures.

The support of flexible mapping comprises these integration scenarios:

- Convert concept component to item;
- Copy item to concept component; and,
- Display item data in concept structure.

Execute the following administrative tasks.

1. Navigate to the Product Development work area (logged in as an administrator). Open the side panel and, in Settings, click Manage Configurations. From the Settings tab, set up the Default Organization: this will be the item organization when you convert concept components to items in PD. Be sure to select the Default Organization.

2. Define an item class for each different type of component. In Innovation Management, open Setup and Maintenance. Navigate to Manage Target System. Create a new PD connector by duplicating it from the Oracle Template ORA_PD.

3. Go to Manage Mapping to External System and click the Add + button to add a subentity, that is, a subclass of the Item class. Select the new item class and add class-specific extensible flex-fields (EFFs).

   ✍ Note: Add EFF attributes on the same hierarchy level of the item class on which the EFF is defined. Also, remember that only one connector can be active at a time.

4. In Innovation Management, go to Manage Mapping where you enable class-specific mapping for the mapping sets Create Item (CDMCreateItem), Copy Item (CDMCopyItem) and Link Item (CDMLinkItem). Select the Mapping Set to which you want to add the class mapping. Then select the entity Item and add the subclass you created.

5. Select the additional item class and add the IM attribute to Item EFF mapping (in the Attribute section). You have created a user-defined attribute for a concept structure in IM, created an EFF in an item structure in PD, and now the data can go from the concept structure to the item structure. Repeat these steps for every type/class combination to be mapped.
Configure Innovation Management for Integration with External Systems

Note: Since a parent class must be added to the mapping before its child classes can be mapped, you may add dummy mappings for intermediate classes if they don’t correspond to a concept component type. In this case, the user can type in any value for the IM Entity Name column.

Oracle Project Portfolio Management Cloud in Innovation Management

Manage Oracle Innovation Management projects by associating work items such as concepts, proposals, and requirements specifications to a project task. You can define rules to associate work items to project tasks and determine the completion of tasks based on work item statuses.

When you associate Oracle PLM objects with tasks in the Project Management work area, you can view the relationships in Oracle PLM pages also. Alternatively, you can search for project tasks in Oracle PLM, and associate them with PLM objects as relationships.

Implement Oracle Innovation Management and Oracle Project Portfolio Management Cloud for Integration

You must implement the following tasks in the Product Requirements and Ideation Management or Concept Design Management functional areas in the Product Management offering to integrate Oracle Innovation Management with Oracle Project Portfolio Management Cloud.

The following table lists the functional areas and their associated tasks.

<table>
<thead>
<tr>
<th>Functional Area</th>
<th>Enable for Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept Design Management</td>
<td>Manage Product Concept Classes</td>
</tr>
<tr>
<td>Concept Design Management</td>
<td>Manage Product Concept Component Classes</td>
</tr>
<tr>
<td>Concept Design Management</td>
<td>Manage Product Concept Statuses</td>
</tr>
<tr>
<td>Concept Design Management</td>
<td>Manage Proposal Statuses</td>
</tr>
<tr>
<td>Concept Design Management</td>
<td>Manage Product Concept Lookups</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Manage Product Idea Classes</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Manage Product Idea Statuses</td>
</tr>
<tr>
<td>Product Requirements and Ideation Management</td>
<td>Manage Product Requirement Classes</td>
</tr>
</tbody>
</table>
### Oracle Innovation Management Business Objects in Project Tasks

The user can manage product-development projects in Oracle Project Management Cloud, but only if the administrator has assigned the user with appropriate job roles, such as Project Manager or Team Member of projects, in the project plan.

Project managers who are assigned Product Manager, Product Design Manager, or Product Portfolio Manager, can perform the following actions:

- Open and manage project work items in Oracle PLM.
- Navigate to Projects from the Relationships tab in Oracle PLM, and view summary information of the related object on hover.
- Define rules to specify statuses that determine when work items can be considered complete. When a work item reaches the appropriate status, the task is updated to complete.
- Set task completion rules, based on status, for each Oracle PLM business object that is associated with a project task.

### Related Topics

- Manage Product Development Projects
- Work Items

## Assign Objects to Project Work Items

### Video

**Watch:** This video demonstrates how to assign an object to a project work item. The content of this video is also covered in text topics.

### Procedure

In this procedure you can see how you assign objects to Project Work Items.

1. Navigate to the Project Management work area.
2. Create a project or open an existing project.
3. Click the **Create Task** icon (+) and add a task to the project.
4. Name the task and click the **Edit** button in the Work Items column.
5. In the Manage Work Items dialog box, select the object type from the **Type** list.
6. Search for and select the object from the Name list.
7. Select a status for the task from the Task Completion Event menu (for a requirements specification) - Submitted, Approved, or Converted. The task completion event menu varies for different objects.
8. Click Save and Close.
   Now that you have created a task and assigned it to the work item, click Save to save the changes.

Troubleshooting

Validate Configurations in Innovation Management

Validate external PLM system configurations and Innovation Management integration with the use cases listed in this section.

- Search for PLM items, documents, and designs
- Read attribute values of PLM items and designs
- Read structure and relationships of PLM items and designs
- Create PLM items and designs
- Create structure and relationships for items and designs
- Read and write file contents to and from file servers

If the following issues occur, refer to the chapter SmartRules in the Agile PLM Administrator Guide for more information:

- Converting to an item structure when the parent item has a file attachment does not create a child item, if the Agile PLM SmartRule is set to either “Copy with Warning” or “Reference with Warning”.
- The Design table has no rows, but the header has a count for users with a default privilege in Agile PLM.
- An Agile PLM Integration Framework error occurs when trying to add some single-list and multi-list attributes.

Related Topics
- Overview on Configuring Agile PLM

Errors in Integrating Innovation Management

This topic lists error messages that you may encounter while integrating Oracle Innovation Management Cloud to external PLM systems, their causes, and actions that may resolve these errors.

The following table lists errors and their possible causes.

<table>
<thead>
<tr>
<th>Error</th>
<th>Cause and Action</th>
</tr>
</thead>
</table>
| **Search errors for Agile PLM items in the Edit Concept page** | **Cause**: Change the Web Services policy assignment to: oracle/wss_saml_bearer_or_username_token_service_policy  
1. Select OWSM > Keystore  
2. Select Manage. |
Error | Cause and Action
---|---
3. | Select **Import**.
4. | Select **Import as a trusted certificate**.
5. | Browse to locate the certificate file. Open the certificate file; or, paste the certificate text contents into the space provided.
6. | Click **OK**.

Repeat for all certificates in the hierarchy up to the Root Certificate Authority.

**SOAP Fault code: MustUnderstand**

**Cause**: The Policy Header sent by the client in the SOAP message was not understood by the Agile PLM server. An immediate child element of the Header element, with the `mustUnderstand` attribute set to 1, was not understood.

**Action**: Confirm that the alias used to import the certificate matches the Keystore.Recipient.Alias value.

**General security error from WSSEcurityEngine: Callback supplied no password for null**

**Cause**: The alias that was used to import the Agile PLM certificate into the Oracle Innovation Management Cloud server keystore does not match the value that was configured for Keystore.Recipient.Alias of the WebServices policy.

**Action**: The alias used to import the Agile PLM certificate into the Oracle Innovation Management server keystore does not match the value that was configured for the Keystore.Recipient.Alias of the WebServices policy.

**Security error**

**Cause**: The server was not able to process the security token; or, the security token failed validation.

**Action**: Check the log files for security-related errors. It is possible that the security certificate was not imported properly as a trusted certificate; or, within the certification hierarchy, the certification authority from Agile PLM is not trusted. All the certificates in the hierarchy up to the Root Certificate Authority must be imported as trusted certificates.

**The thumbnail image for an Agile PLM item does not display when the item is added to a concept structure**

**Cause**: Thumbnails are not configured in Agile PLM or enabled for the current user.

**Action**: Confirm that the user account in Oracle Innovation Management Cloud has an identical user account in Agile PLM. Thumbnail settings must be enabled for both Agile PLM and the Agile PLM user account.

The Oracle Innovation Management Web Service Client sends a SOAP Message to the Agile PLM Server, which is not processed on the server side.

- **Agile Java Client > Server Settings > Preferences > Thumbnail Display > Enable**
- **Agile Java Client > Users > [user account] > Preferences > Thumbnails: On**

Ensure that each file type used is enabled for AutoVue in Agile PLM.

- **Agile Java Client > System Settings > Viewers and Files > File Association [tab]**

**An Error occurred in the Applications Unlimited PLM Integration Framework.** For example, Item<item name>.Attribute<attribute ID> not found.

**Cause**: The requested attribute is not enabled.

**Action**: Confirm that every Agile PLM attribute intended to be mapped to a Oracle Innovation Management Cloud attribute has been enabled (for example, Visible is set to Yes). Resolve the issue and try again.
Why does the web service connection fail?

A user of an Innovation Management application may try to call a web service, and the call fails. The web service call may fail due to several possible exceptions, including path certification, bad encryption, and policy enforcement exceptions. The end-user should create a service request for the administrator to resolve the possible issues. This summary may assist both the user and the administrator.

Verify that these prerequisite steps are completed:

1. Obtain details of the WSDL URL and the user credentials to use from the web service provider.
2. Get the server encryption certificate and the Certificate Authority (issuer) certificate from the web service provider.
3. Create a user-defined field for an object that has a calculated default value.
4. Prepare the Groovy script for the expression used to calculate the field’s default value. The Groovy code must prepare the argument values, which in this example are two values that are summed.

To call a web service from a Groovy script that is secured with message protection, verify that these tasks are completed:

1. Create the web service connection.
2. Add the web service call to the Groovy script, and check whether the call succeeds.
3. Contact the administrator to resolve run time exceptions

Create a service request for your administrator:

   a. Retrieve the server encryption certificate and the Certificate Authority (issuer) certificate from the web service provider.
   b. Attach the server encryption certificate and the issuer certificate to the service request, and include the WSDL location, and error details.
   c. Submit the service request.

   The administrator adds the server encryption certificate and the issuer certificate into the Oracle Innovation Management trust store. The administrator also creates an alias for the server encryption key, which you must use to recreate the web service connection.

   d. Wait until your administrator informs you that the certificates have been imported, and that the server encryption alias has been created; then close the service request.

4. Recreate the web service connection.
5. Verify that the web service call succeeds.

Related Topics

- Overview of the Oracle Social Network Objects in Oracle Innovation Management
- Enable Single Sign-On for Innovation Management
- What are the prerequisites for Oracle Social Network integration
## Appendix: Configure Agile PLM for Integration with Innovation Management

### Overview on Configuring Agile PLM

This topic details the process of configuring Agile Product Lifecycle Management (PLM) to integrate with Oracle Innovation Management Cloud.

### Prerequisites

Ensure that the configuration tasks are executed by an administrator with knowledge of both:

- Oracle Agile PLM Java Client, and
- Oracle Functional Setup Manager (FSM).

The administrator user must have all required Administrator-related privileges in Agile PLM.

These Agile PLM versions can be integrated with Oracle Innovation Management Cloud:

- **Oracle Agile PLM 9.3.6** or any Release Update Pack installed
  
  OR

- **Oracle Agile PLM 9.3.5** with Hot Fixes 15 and 24 installed, or any Release Update Pack installed
  
  OR

- **Oracle Agile PLM 9.3.4** with Hot Fixes 1, 2, 3, 11, 26, 43, 52, and 146 installed

as per the guidelines in the Agile Installation documentation.

You must also complete the **Agile PLM 9.x ("A9") and File Manager Web Services Setup Checklist** to configure web service security.

### Configuration Process

This table lists required configuration tasks, which must be completed in the numbered sequence.

<table>
<thead>
<tr>
<th>Task</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Create a service request to exchange security certificate information.</td>
<td>Exchange security certificates between the Agile PLM and Oracle Innovation Management systems.</td>
</tr>
<tr>
<td>Task</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2. Import the Innovation Management security certificate to Agile PLM.</td>
<td>Use Enterprise Manager to import the security certificate that you receive from Support.</td>
</tr>
<tr>
<td>4. Make sure the endpoint is visible from outside your firewall.</td>
<td>Confirm that the endpoint is accessible.</td>
</tr>
<tr>
<td>5. Create Oracle Innovation Management Cloud users in Agile PLM.</td>
<td>Create users in Agile PLM with user IDs identical to those in Oracle Innovation Management.</td>
</tr>
</tbody>
</table>
| Alternatively, configure Agile PLM to use the same LDAP server as Oracle Innovation Management. | Set up Oracle Identity and Access Management.  
Refer to the Oracle Agile PLM Administrator Guide. |
| 7. Add Oracle Innovation Management attributes to required privileges in Agile PLM. | Enable appropriate roles for Oracle Innovation Management users in Agile PLM.  
Enable **Read Items**, **Modify Preliminary Items**, and **Modify Released Item** privileges for Oracle Innovation Management users in Agile PLM. |
| 8. Configure Reference Objects in Agile PLM.                        | Create reference objects in Agile PLM to support linking back from Agile PLM to Oracle Innovation Management. |
| 10. Configure and activate the connector.                            | Verify the connector can communicate between the two products. |

## Configure Message Protection for Agile PLM

For the integration between Oracle Innovation Management Cloud and Agile PLM to work, you must configure a domain trust between the WebLogic domain, where Oracle Innovation Management is running, and the domain where Agile PLM is running. 

In production environments, the security certificates that are configured in the identity keystore of an Agile PLM WebLogic domain are official certificates provided by a Certificate Authority (CA) to ensure that the identity of the Agile PLM host is officially certified.
Even so, the certificate provided by the CA and imported into the identity keystore may not be sufficient for the Cloud application domain to trust the identity of the target system host, given that the hierarchy of certification up to the root certificate may not be known to the WebLogic domain.

To resolve this potential problem, you can permit identity trust by:

- providing the public certificate that is associated with your system host and all certificates in the hierarchy to the root; and,
- submitting a service request to ask for those certificates to be imported to the trust store of the WebLogic domain.

Follow these steps to configure message protection for Agile PLM:

1. Create a service request with the following information:
   a. Name of the service request: Message Protection Configuration on Oracle Cloud To Integrate With Agile PLM
   b. Include the following information in the comments section:
      • Specify the version of Agile PLM you are using.
      • Specify message protection policy to oracle/wss11_saml_token_bearer_over_ssl_client_policy.
   c. Submit the service request.
   d. Wait until your administrator informs you that the service request has been processed.

2. Import the signature certificates used by Oracle Cloud into Agile PLM.

When the service request is processed, the security certificates used by the Oracle Cloud WebLogic domain to sign the Web Service requests are attached to the service request. You are notified that you can proceed to import the certificate to the trust keystore of Agile PLM.

Download the certificates attached to the service request. The trusted certificates must be imported to the Oracle Web Services Manager KSS keystore and the full certificate chain must be available.

   a. Open Enterprise Manager for your Agile PLM installation.
   c. Expand OWSM > Select the Manage button.
   d. Click the Import button, a pop-up window appears.
   e. Select Trusted Certificate for Certificate Type, provide an alias, and copy and paste or choose a local file containing the certificate.
   f. Repeat steps 3.4 and 3.5 to import each certificate attached in the service request.

3. Determine the target URL for the Web Services.

Determine the target URL for the Web Services that are being used for the integration. Enter the corresponding values in Oracle Cloud using Setup and Maintenance. The typical endpoint for Agile PLM is as follows:

   `{protocol}://{host}:{port}/CoreService/services/{service}?wsdl`; for example, `http://example.com:7001/CoreService/services/Search?wsdl`

   a. Enter the values for protocol, host, port, and context root into the corresponding fields for registering the target endpoint in Oracle Cloud, using the Setup and Maintenance workspace, as follows:
      • Sign in to Oracle Cloud as a user who has the privilege to modify configuration values, and then click Setup and Maintenance.
      • On the Setup and Maintenance page, search for the Register Agile PLM task.
      • Click the task name.
• Select the **Server Protocol** and enter values for External Server Host and External Server Port in the **Server Details** section. Enter the Context Root in the row named **AgileA9CoreServices** in the **Associated Modules** section.

• Click **Save and Close**.

4. **Make sure the endpoint is visible from outside your corporate firewall.**

   The web services endpoint registered in Oracle Cloud must be reachable from outside your corporate firewall, so Oracle Cloud can call the corresponding web services. Make sure the port is open for incoming traffic and the host name is valid from outside the firewall.

5. **Configure users to ensure that they are consistent on both systems.**

6. **Configure and activate the connector.**

### Configure Users in Agile PLM

After you configure the web services security layer, ensure that the user ID defined in Oracle Innovation Management also exists in Agile PLM. The user must also have enough privileges granted to execute required web services.

The identity of the user signing in to Oracle Innovation Management is applied to the web service call; it’s implied that the same user attempts to execute actions in Agile PLM through the web services interface.

Use the following methods to make the required users available in Agile PLM:

- **Method 1:** Create the same Oracle Innovation Management users in Agile PLM using the Java Client.
- **Method 2:** Configure Agile PLM to use the same LDAP server as Oracle Innovation Management.

### Innovation Management Attributes in Agile PLM (On Page Two)

Enable users to add attributes, and add or edit values of attributes on items in Agile PLM, that can be used to hold information related to Oracle Innovation Management.

In Agile PLM Java Client, enable the **Page Two** tab for **Parts** and the attributes listed in the following table.

<table>
<thead>
<tr>
<th>Attribute API Name</th>
<th>Attribute Name</th>
<th>Type</th>
<th>Attribute Base ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDate16</td>
<td>Last Cost Update</td>
<td>Date</td>
<td>2000019547</td>
</tr>
<tr>
<td>IMList26</td>
<td>Country of Origin</td>
<td>List</td>
<td>2000019548</td>
</tr>
<tr>
<td>IMMoney11</td>
<td>Material Cost</td>
<td>Money</td>
<td>2000019549</td>
</tr>
<tr>
<td>IMMoney12</td>
<td>Nonmaterial Cost</td>
<td>Money</td>
<td>2000019550</td>
</tr>
<tr>
<td>IMMoney13</td>
<td>Nonrecurring Cost</td>
<td>Money</td>
<td>2000019551</td>
</tr>
</tbody>
</table>
### Add Oracle Innovation Management Attributes to Required Privileges in Agile PLM

Firstly, enable the attributes that have API names starting with `IM` in Agile PLM (on Page Two). Next, add Oracle Innovation Management attributes to the required Agile PLM privileges to ensure successful integration.

The privileges in Agile PLM are:

- Read Items
- Modify Preliminary Items
- Modify Released Items

Execute the following steps:

1. In the Agile PLM Java Client Admin tab, expand User Settings, then expand Privileges and double-click All Privileges.
2. Repeat the following steps for each of the privileges.
   
   a. Search for the privilege in the All Privileges window and double-click it.
   b. Expand the menu of the Applied to field and ensure all the attributes are present in the Selected panel. Move them from the list, if needed.

### Configure Reference Objects in Agile PLM

To support linking back from Agile PLM to the Oracle Innovation Management system, the integration engages the External References functionality in Agile PLM.

To configure this functionality:

1. Sign in to the Agile PLM Java Client with Administrator privileges.
2. Select the Admin tab and expand System Settings.
3. Expand the node **Reference Objects Management** and double-click **Applications**. The **Applications** window is displayed.

4. To add the Oracle Innovation Management system as an additional Application, click **New** in the **Applications** window.

5. Complete the fields to create the application. While **Name**, **API Name**, and **Description** can be freely defined, pay attention to the following fields:
   - **Host Base URL** must include the protocol, but no context path or port. Example: https://sample.us.oraclecloud.com
   - **Port** must include the port number (digits only) of the port where the Oracle Innovation Management application is running. Example: 443

   **Note:** The host base URL is the one you used to sign Oracle Innovation Management Cloud with no context path and port. For on-premise deployments, sign in to your Innovation Management system and navigate to **Concept Design**. Use the URL in the browser to determine the protocol, host name, and port.

   - **Virtual Path** must remain blank.
   - **User ID and Password** may remain blank
   - **Enabled** must be set to **Yes**.

6. Click **OK** to create the application.

7. In the **Admin** tab, expand **Data Settings** and double-click **Classes** underneath it.

   The **Classes** window appears.

8. Scroll to the class **Reference objects** (note that there is a base class called **Reference Objects**; select the class which is directly following the base class).

9. Click **New** to create a subclass of the class **Reference Objects**.

10. Complete the fields to create a subclass.

11. Click **OK** to create the subclass.

12. In the window that opens, complete the following fields:

   - **Application**: Select the application from the menu, which is the Reference Object created in step 6.
   - **Available On**: Expand the menu from the list, select **Items** and move it to the **Selected** panel.

   Click **OK**.

13. Click **Save**.

### Privileges for Innovation Management Users in Agile PLM

In Agile PLM, enable the required privileges for Oracle Innovation Management users so they can interact with Agile PLM Reference Objects successfully.

1. Sign in to the Agile PLM Java Client with Administrator privileges.
2. Select the **Admin** tab and expand **User Settings**, then expand **Privileges**, and double-click **All Privileges**.
3. In the **All Privileges** window, click **New** to create a privilege.
4. Name the privilege **Discover Reference Objects**; optionally, provide a **Description**.
5. In the **Privilege** menu, select **Discovery**.
6. Click **New** next to the **Criteria** menu, to create a **Criteria**.
7. Name the criteria **All Reference Objects**; optionally, provide a **Description**.
8. In the **Object Type** field, select **Reference Objects**.
9. Click **OK** to create the criteria.
10. Click **OK** to create the privilege.
11. In the **All Privileges** window, click **New** to create a privilege.
12. Name the privilege **Read Reference Objects**; optionally, provide a **Description**.
13. In the **Privilege** menu, select **Read**.
14. In the **Criteria** menu, select the newly created criteria.
15. Expand the menu in the **Applied to** field and move all the fields to the **Selected** panel. Click **OK**.
16. Click **OK** to create the privilege.
17. Assign the two created privileges to the appropriate roles according to the users configured in Oracle Innovation Management.
work item
An item that represents a unit of work that team members are performing on a project task. The status of the work item can determine if a task is complete.