Oracle Public Sector Compliance and Regulation Cloud

Implementing Your Cloud Integrations

20C

Contents

Preface

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

Using Oracle Applications

This topic explains the text conventions used in this guide and points you to where you can find more information about using Oracle applications.

Conventions

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

Convention	Meaning
boldface	Boldface type indicates user interface elements, navigation paths, or values you enter or select.
monospace	Monospace type indicates file, folder, and directory names, code examples, commands, and URLs.
>	Greater than symbol separates elements in a navigation path.

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.*

Documentation Accessibility

This topic covers accessibility concepts for this guide.

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

This topic explains how to contact Oracle for support and to provide feedback.

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 Public Sector Compliance and Regulation applications help and guides! You can send an e-mail to: *PSCR_US@oracle.com*.

Oracle Public Sector Compliance and Regulation Cloud
Implementing Your Cloud Integrations

20C

F29834-01

1 Setting Up Workflow

Workflow Basics

You define your transaction workflow using the Process Builder in Oracle Autonomous Integration Cloud (OIC). This topic provides a general introduction to some important OIC terms and lists the high-level steps for setting up workflow for Public Sector Compliance and Regulation transactions. The list of steps includes links to additional specific information for this solution. Use this specific documentation in conjunction with the OIC documentation to learn how to set up workflow for Public Sector Compliance and Regulation.

Workflow is supported for these offerings:

- Permits
- Planning and Zoning
- Business Licenses

Note: The Code Enforcement offering does not utilize workflow currently.

To familiarize yourself with the Process Builder in OIC, see your OIC documentation at:*https://docs.oracle.com/en/cloud/paas/integration-cloud/user-processes/develop-structured-processes.html*.

Note: Oracle provides a Solution Package with sample workflow configurations. You can clone these samples and use them as starting points to create your own workflow. You can also set up your workflow from scratch.

Note: Currently, in the context of data object parameters, data association parameters, and REST resource attributes, the terms *record, transaction,* and *permit* are interchangeable.

Important OIC Terms

The OIC object where you set up your workflow tasks is called a *process definition.*

The following table describes the hierarchy of objects for a process definition. When you set up a type and you choose the appropriate process definition, you need to specify each of these hierarchical objects.

Object	Description
Space	Spaces are an organizational tool similar to a folder. Your agency chooses the spaces that make sense for your organization. For example, you can create separate spaces for different categories of s.
Application	Applications are functional areas within Spaces. Within an application, you can access a variety of features, including processes (workflow) and integrations.

Object	Description
	Certain configurations, including integrations and roles, are defined at the application level and shared by all of the application's process definitions. Therefore, you can simplify the setup process by grouping related process definitions into a single application.
Version	When you activate a modified application to make it available for use, you choose a version number to assign. New and modified process definitions can't be associated with a transaction type until you activate a version of the application that includes your changes. **CAUTION:** If you reuse the same version number when you activate an application, all open process instances using that version are terminated. To prevent this, use a new version number and then update any transaction types that need to use the new version.
Process Definition	A process definition is a specific workflow process. When different types have the same workflow, they can use the same process definition. See *Reviewing a Sample Process Definition* to walk through an example of a process definition for workflow.

High-Level Steps for Setting Up Workflow

The OIC documentation provides complete information on setting up workflow, but these are the high level steps, with some notes about product-specific considerations:

1. Set up a proxy role and user for accessing Oracle Integration Cloud.

   See *Setting Up a Proxy Role and User for Oracle Integration Cloud*.
2. Create the OIC space and application for your workflow.
3. Set up your transaction-specific integrations.

   See *Setting Up the Communications Connector* and *Setting Up the Transactions Connector*.
4. Create the process definition.

   Note: For Public Sector Compliance and Regulation workflow, you must create the process definition using the type *Message.*
5. Set up swimlanes.

   Swimlanes are equivalent to roles in the Public Sector system.
6. Design your process flow, which includes start and end events, human tasks, system tasks, gateway decision points, and arrows that define the flow through these objects.

   Your OIC documentation explains how to create a process flow, but there are additional considerations for Public Sector Compliance and Regulation. See *Setting Up Process Definitions for Workflow* for these important product-specific tasks:

   - Setting Up Transaction Data Definitions for a Process

- Defining Arguments for the Start Event
- Defining Data Associations for the Start Event
- Defining Data Associations for Sending Notifications
- Defining Data Associations for Sending Transaction Status Updates
- Defining Data Associations for Retrieving Transaction Base Data
- Defining Data Associations for Retrieving Transaction Field Data
- Defining Data Associations for Retrieving Transaction Type Data
- Defining Statuses (Outcomes) for Human Tasks
- Defining Conditional Logic for Gateways

7. Use custom properties to add specific information to the human tasks in your workflow.

   See *Using Custom Properties*.
8. Activate your application and assign it a version number.

   Activating an application makes its new and modified process definitions available to associate with a transaction type. If you reuse the same version number when you activate the application, all open process instances using that version are terminated. To prevent this, use a new version number and update any impacted types so that they reference the new version number.

   > **Note:** For applications used for the Planning and Zoning solution, activate the application with the Use Fault Policies option unchecked on the Activation Options page.
9. If this is the first time that the application has been activated, use the Manage Roles functionality in OIC to map swimlanes to roles.

   Swimlanes cannot be mapped until the application has been activated. The mapping applies to all process definitions in the application.

   See *Mapping Workflow Swimlanes to Roles*.

Managing Intersystem Connection Disruptions

There is a built-in mechanism for handling temporary unavailability of OIC. An automatic synchronization process runs every hour and scans a database table containing relevant information pertaining to any transaction, such as a transaction in the state of *pending submittal.* When the process discovers items in the table, it reconnects to OIC to retrieve the process instance for that transaction. After ten attempts, running once each hour, if reconnecting to OIC is not successful, the transaction becomes stale and manual intervention is required.

For example, if a registered user submits an application when OIC happens to be unavailable, that gets set to a state of pending submittal because no process instance can be associated with it due to OIC being unavailable. In that case, the system stores the information for that application in a specific database table, which will be discovered by the synchronization process within an hour. Assuming OIC is available, the synchronization process retrieves the process instance from OIC, associates it with the application, and sets the transaction status to *Submitted.*

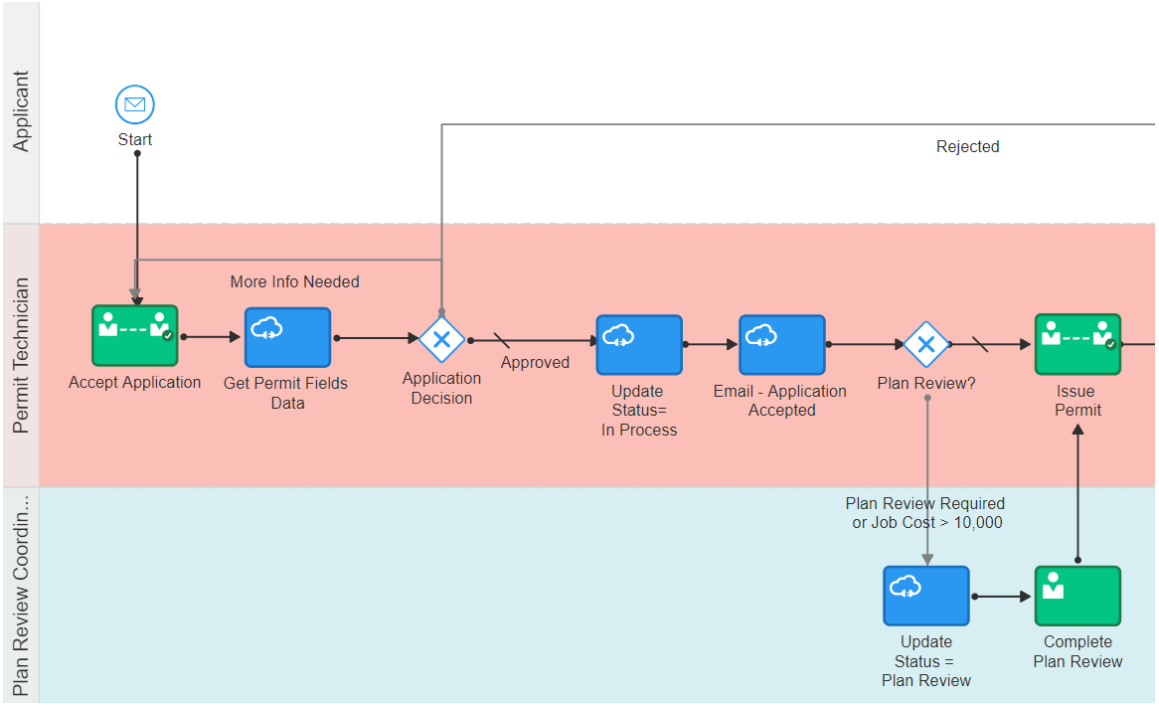
Reviewing a Sample Process Definition

A process definition provides a defined flow for processes such as the transaction lifecycle of a permit. This flow can include system tasks, human tasks, and decision gateways. You define your flow using the Process feature in Oracle Autonomous Integration Cloud (OIC). The Process feature provides a visual design environment to help you create easily understood workflow process definitions.

> **Note:** Currently, in the context of data object parameters, data association parameters, and REST resource attributes, the terms *record, transaction,* and *permit* are interchangeable.

Let's look a a sample process definition for a building permit.

This image shows the first half of the sample process, from the time the permit is submitted until it is issued.



The following table identifies the types of objects shown in the illustration:

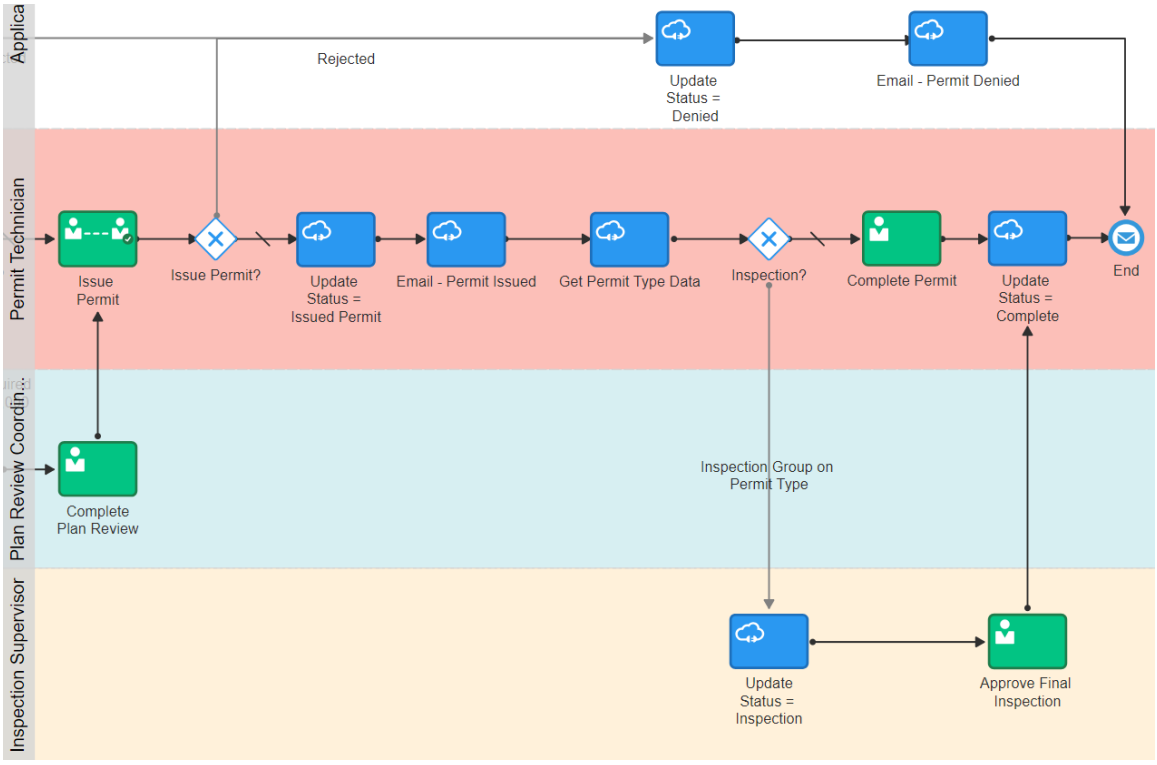
Object	Description
Swimlanes	Horizontal bands in the process map represent the roles involved in the process.
Start and End Events	All paths through the workflow process must begin at the Start event and finish at the End event.
Human tasks	Green boxes with an image of a person represent tasks that are performed by humans.
System tasks	Blue boxes with an image of a cloud represent tasks that the system performs.

Object	Description
Gateways	White diamonds represent decision points, where the process flow can branch based on criteria you define.
Arrows	One-directional arrows define flows through the process. Gateways are the only objects that have multiple exit arrows. The exit arrow with a slash through it represents the default option after a gateway. All other exit arrows contain business logic for defining the conditions when the arrow is used.

With these definitions in mind, let's look at the sample process flow:

1. **Start:** The process starts when a permit application is submitted, which sends a message to OIC to instantiate the workflow process.
2. **Accept Application:** A human performs the task of accepting the application and selecting a task status that represents the task outcome.
3. **Get Permit Fields Data:** This system task retrieves permit field data to be used later in the process, when it's time to determine whether a plan review is required.
4. **Application Decision:** Exit arrows from this gateway determine the next step based on the outcome of the Accept Application human task.
    a. If more information is needed, the application acceptance task is reinstantiated. This loop continues until the task has a different outcome.
    b. If the application is rejected, a system task updates the permit status to *Denied,* then another system task sends the applicant an email notification that the permit was denied, then the process ends.
    c. If the outcome is anything else, the process continues.
5. **Update Status = In Process:** This system task updates the permit status to *In Process.*
6. **Email - Application Accepted:** This system task notifies the applicant that the permit was accepted.
7. **Plan Review:** Exit arrows from this gateway determine the next step based on the outcome of the Accept Application task and based on the job cost that was retrieved by the Get Permit Fields Data task:
    a. If the Accept Application outcome indicates that a plan review is required, or if the job cost is greater than 10,000, the **Update Status = Plan Review** system task updates the permit status to *Plan Review,* then a human completes the **Complete Plan Review** human task. When the Complete Plan Review is complete, the process continues.
    b. If a plan review is not required, the process continues.
8. **Issue Permit:** A human performs the task of issuing the permit and enters a task status that represents the task outcome (whether the permit was issued or rejected).

The following image shows the remainder of the sample workflow, after a human completes the Issue Permit task.



These steps describe the remainder of the workflow process, after the human task for issuing a permit:

1. **Issue Permit:** Exit arrows from this gateway determine the next step based on the outcome of the task for issuing a permit:
    a. If the permit is rejected, the **Update Status = Denied** This system task updates the permit status to *Denied,* then the **Email - Permit Denied** system task notifies the applicant that the permit was denied, then the process ends.
    b. If the outcome is anything else, the process continues.
2. **Update Status = Issued Permit:** This system task updates the permit status to *Issued Permit.*
3. **Email - Permit Issued:** This system task notifies the applicant that the permit was issued.
4. **Get Permit Type Data:** This system task retrieves permit type information for use in determining whether an inspection is needed.
5. **Inspection:** Exit arrows from this gateway determine whether an inspection is needed:
    a. If the permit type includes an inspection group, the **Update Status = Inspection** system task updates the permit status to *Inspection.* A human then completes the **Approve Final Inspection** task and enters the task outcome. The process then continues.
    b. If an inspection is not required, a human performs the **Complete Permit** task and enters the task outcome. The process then continues.
6. **Update Status = Complete:** this system task updates the permit status to *Complete.*
7. The process ends.

Setting Up a Proxy Role and User for Oracle Integration Cloud

Oracle Autonomous Integration Cloud (OIC) provides the tools for setting up workflow processes. This topic provides information about using the Security Console to set up a proxy user that the Public Sector system uses to access OIC.

In this procedure, create a user and assign the PSCR Proxy User for OIC (CUSTOM_PSCR_OIC_PROXY_USER) role to that user. You use the Security Console to complete this task. This user is the OIC proxy user that the OIC system uses to connect to Public Sector Compliance and Regulation to exchange data during transaction processing.

For more information about using the Security Console, see: *Using the Security Console*.

To create the OIC proxy user:

1. Navigate to the Security Console.

   To navigate to the Security Console, you have these options:

   - In Functional Setup Manager, click the task: *Create Process Cloud Service Proxy User.*
   - Click Setup and Maintenance on the Agency Springboard, and on the Fusion Applications home page, select **Navigator** > **Tools** > **Security Console.**
2. Click the Users tab.
3. On the Use Accounts page, click **Add User Account.**
4. On the Add User Account page in the User Information section, enter a **Last Name** and **User Name** of your choice.

   > **Note:** The name given for the proxy user will be displayed on the Status History page of the transaction detail pages, for permits, planning applications, and so on. As such, you may want to use a generic name, such as *System, Workflow,* or something similar.
5. Enter a **Password** of your choice and confirm it.
6. Click **Add Role** for the Roles grid, and assign this role to your proxy user:

   - Role Name: *PSCR Proxy User for OIC*
   - Role Code: *CUSTOM_PSCR_OIC_PROXY_USER*
7. Click **Save and Close.**

Note: You will add this proxy user to OIC process definitions.

Setting Up the Communications Connector

The communications connector enables OIC to send data to the communications center in the Oracle Public Sector Compliance and Regulation system using a POST operation. This connector is used when a workflow process definition includes a communication task, such as sending a permit applicant an email when the permit status changes.

Oracle provides a Solution Package with sample integration configurations. You can clone these samples and use them as starting points for your own connectors. The instructions in this procedure explain how to set up the communications connector from scratch.

The following procedure explains how to set up the communications connector with the specific integration information that is required by the Public Sector system. For general instructions related to setting up integrations in OIC, refer to your OIC documentation at *https://docs.oracle.com/en/cloud/paas/integration-cloud/user-processes/integrate-applications-and-services.html*

> **Note:** Before you set up the communications connector, you must create the Space and the Application for your workflow processes. See *Workflow Basics*.

> **Note:** Currently, in the context of data object parameters, data association parameters, and REST resource attributes, the terms *record, transaction,* and *permit* are interchangeable.

To set up the communication connector:

1. Go to My Oracle Support, access Doc ID 2449735.1, *Public Sector Compliance and Regulation: JSON Files for Transaction Integration,* and download the following files that you will use later in this procedure:
    - RequestCommunications.json
    - ResponseCommunications.json
2. Access the main console in OIC.
3. In the list of OIC applications, click the application with your transaction workflow.
4. Click the **Integrations** option in the left frame.
5. Click the **Create** button, then in the pop-up menu under the Create button, select **External** > **REST**
6. In the Create REST Connector window, enter the following:

Page Element	Description
Name	Enter a descriptive name such as *CommunicationsConnector.* **Note:** The name *CommunicationsConnector* is suggested, however, you can choose your own name if needed.
Base URL	Enter the URL for your Oracle Public Sector Cloud REST API resources. The URL follows this pattern, where *ServerName* is the server name for your instance of the application: https://*ServerName*/fscmRestApi/resources/11.13.18.05
Open Immediately	Select this check box if it is not already selected.

7. Click **Create.**
   The Rest Connector Editor opens.

8. To set up security for this integration, click the **Edit** link for the Configuration section.

   > **Note:** If you prefer to set up security when you activate the workflow application, you can skip the security-related steps in this procedure and skip ahead to step 13. Setting up security now simplifies the application activation steps.

9. Click the **Security** tab.
10. In the **Security Type** field, select *APP Id - Basic Authentication.*
11. Complete these additional fields that appear after you select the **Security Type:**

Page Element	Description
Keystore Credential	If you previously created a keystore credential, select it. Otherwise, leave this field set to *[New Key]* so that the system will create the keystore credential when you apply your changes.
Key Name	If you selected *[New Key]* as the keystore credential, enter the name to give to the new keystore. If you selected an existing keystore credential, this field is read-only and displays the key name.
Username	Enter the user name for the process cloud proxy user that you previously created. If you're using an existing keystore credential, that credential supplies a default username.
Password	Enter the password for the process cloud proxy user that you previously created. If you're using an existing keystore credential, that credential supplies a default password.

12. Click **Apply** to save the security information and close the Configuration section.
13. In the Resources section of the Rest Connector Editor, click **Add.**
14. Expand the new Resource section that appears, and enter the following values:

Field	Value
Name	*OutboundCommunications*
Path	*publicSectorCommunicationRequests* When added to the base URL, this completes the path to the communications-related REST APIs.

15. In the **Operations** section, click the **Add** button and then select **POST operation** from the drop-down menu.

16. Click the new **POST** operation.
17. Enter *Trigger transaction communications* in the **Documentation** field.

    You can leave the default values in the other fields, including leaving the **Path** field blank.
18. Click **Request**
19. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
20. Enter *RequestCommunications* in the **Name** field.
21. Click **Schema.**
22.  Click the **Import from File** icon next to the **Schema** button.
23. Locate and upload the *RequestCommunications.json* file that you downloaded from My Oracle Support.

    The imported JSON code appears in the Import Business Object from JSON window.
24. Click the **Import** button at the bottom of the window to save the code and close the window.
25. Ensure that the following values now appear for the POST operation request:

Page Element	Value
Body	*BusinessData.RequestCommunications*
Media Type	*Custom*
Media Type details	*application/vnd.oracle.adf.resourceitem+json*

26. Click **Response.**
27. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
28. Enter *ResponseCommunications* in the **Name** field.
29. Click **Schema.**
30.  Click the **Import from File** icon next to the **Schema** button.
31. Locate and upload the *ResponseCommunications.json* file that you downloaded from My Oracle Support.

    The imported JSON code appears in the Import Business Object from JSON window.
32. Click the **Import** button at the bottom of the window to save the code and close the window.
33. Ensure that the following values appear for the POST operation response:

Page Element	Value to Enter
Body	*BusinessData.ResponseCommunications*
Media Type	*application/JSON*

34. Click **Apply.**
35. Click **Save.**

Setting Up the Transactions Connector

The transactions connector enables OIC to exchange transaction-related information with the Public Sector Compliance and Regulation system.

Oracle provides a Solution Package with sample integration configurations. You can clone these samples and use them as starting points for your own connectors. The instructions in this procedure explain how to set up the transactions connector from scratch.

The procedures that follow are for entering the specific information that is required by the Public Sector Compliance and Regulation system. For general instructions related to setting up integrations in OIC, refer to your OIC documentation at *https://docs.oracle.com/en/cloud/paas/integration-cloud/user-processes/integrate-applications-and-services.html*

Note: Before you set up connectors, you must create the Space and the Application for your workflow processes. See *Workflow Basics*.

Note: Currently, in the context of data object parameters, data association parameters, and REST resource attributes, the terms *record, transaction,* and *permit* are interchangeable. In some resource attributes, such as paths or parameters, *trans* is used in place of *transaction* for simplicity.

Procedure Overview for Setting Up the Transactions Connector

Setting up the Transactions Connector comprises multiple procedures with multiple steps. The following set of steps outline the high-level set of procedures involved with this task. Each item in the following list is explained in more detail in the following sections, in the listed sequence.

1. Download Required JSON Files for Workflow Integration Configuration
2. Create the Transactions Connector
3. Add the Transaction Resource
4. Add the PATCH Operation for Transaction Statuses
5. Add the GET Operation for Transaction Base Data
6. Add the GET Operation for Transaction Fields Data
7. Add the GET Operation for Transaction Data
8. Add the GET Operation for Transaction Assignee
9. Add the Transaction Type Resource
10. Add the GET Operation for Transaction Type Data

Step 1: Download Required JSON Files for Workflow Integration Configuration

For creating integrations from scratch, Oracle provides a set of JSON files for defining the integration between OIC and the Public Sector Compliance and Regulation system. Download all of the files first so that you can access them easily while completing the procedures documented in this topic.

To download the required JSON:

1. Sign on to My Oracle Support.
2. Access Doc ID 2449735.1, *Public Sector Compliance and Regulation: JSON Files for Transaction Integration.*

3. Download the following files to a local folder:
    - RequestTransStatusUpdate.json
    - ResponseTransStatusUpdate.json
    - ResponseTransBase.json
    - ResponseTransFields.json
    - ResponseTransData.json
    - ResponseTransAssignee.json
    - ResponseTransType.json

Step 2: Creating the Transaction Connector

> **Note:** This procedure explains how to create the transaction connector. Additional procedures that follow this one explain how to set up the operations for this connector.

To set up the transactions connector:

1. Access the main console in OIC.
2. In the list of OIC applications, click the application for your workflow process.
3. Click the **Integrations** option in the left frame.
4. Click the **Create** button, then in the pop-up menu under the Create button, select **External** > **REST**
5. In the Create REST Connector window, enter the following:

Page Element	Description
Name	Enter a descriptive name, such as *TransactionConnector.* **Note:** The name *TransactionConnector* is suggested, however, you can choose your own name if needed. This documentation refers to *TransactionConnector.*
Base URL	Enter the URL for your Oracle Public Sector Cloud REST API resources. The URL follows this pattern, where *ServerName* is the server name for your instance of the application: https://*ServerName*/fscmRestApi/resources/11.13.18.05
Open Immediately	Select this check box if it is not already selected.

6. Click **Create.**

   The Rest Connector Editor opens.
7. If you want to set up security for this integration now, click the **Edit** link for the Configuration section.

> **Note:** If you prefer to set up security later when you activate the workflow application, you can skip the security-related steps in this procedure. Setting up security now simplifies the application activation steps.

8. Click the **Security** tab.
9. In the **Security Type** field, select *APP Id - Basic Authentication.*
10. Complete these additional fields that appear after you select the **Security Type:**

Page Element	Description
Keystore Credential	If you previously created a keystore credential, select it. Otherwise, leave this field set to *[New Key]* so that the system will create the keystore credential when you apply your changes.
Key Name	If you selected *[New Key]* as the keystore credential, enter the name to give to the new keystore. If you selected an existing keystore credential, this field is read-only and displays the key name.
Username	Enter the user name for the process cloud proxy user that you previously created. If you're using an existing keystore credential, that credential supplies a default username.
Password	Enter the password for the process cloud proxy user that you previously created. If you're using an existing keystore credential, that credential supplies a default password.

11. Click **Apply** to save the security information and close the Configuration section.
12. Click **Save**

Step 3: Add the Transactions Resource

> **Note:** Before starting this procedure, be sure to complete the procedure "Setting Up the Transactions Connector."

1. Access the main console in OIC.
2. In the list of OIC applications, click the application for your workflow process.
3. Click the **Integrations** option in the left frame.
4. Click the **TransactionConnector** integration.
5. In the Resources section of the Rest Connector Editor, click **Add.**
6. Expand the new Resource section that appears, and enter *TransactionResource* in the **Name** field.

Step 4: Add the PATCH Operation for Transaction Statuses

Workflow in OIC uses the PATCH operation to update the status of a transaction.

To set up the PATCH operation:

1. In the **Operations** section of TransactionResource, click the **Add** button and then select **PATCH operation** from the drop-down menu.
2. Click the new **PATCH** operation.
3. Enter the following information:

Page Element	Value
Name	*patchTransactionStatus*
Path	*{transResource}/{transRecordKey}* Although you can choose different names for the resource name and record key parameters, this procedure assumes that you use the given values.
Documentation	*Update transaction status.*

4. Click **Request**
5. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
6. Enter *RequestTransStatusUpdate* in the **Name** field.
7. Click **Schema.**
8. Click the **Import from File** icon next to the **Schema** button.
9. Locate and upload the *RequestTransStatusUpdate.json* file that you downloaded from My Oracle Support.
   The imported JSON code appears in the Import Business Object from JSON window.
10. Click the **Import** button at the bottom of the window to save the code and close the window.
11. Ensure that the following values now appear for the PATCH operation request:

Page Element	Value to Enter
Body	*BusinessData.RequestTransStatusUpdate*
Media Type	*Custom*
Media Type details	*application/vnd.oracle.adf.resourceitem+json*

12. In each row of the **Parameters** list, click the *Enter a description* text and enter a description.
    These are example descriptions:

Parameter	Description
transResource	*Transaction Resource Name*

Parameter	Description
transRecordKey	*Transaction Record Key*

13. Click **Response.**
14. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
15. Enter *ResponseTransStatusUpdate* in the **Name** field.
16. Click **Schema.**
17.  Click the **Import from File** icon next to the **Schema** button.
18. Locate and upload the *ResponseTransStatusUpdate.json* file that you downloaded from My Oracle Support.

    The imported JSON code appears in the Import Business Object from JSON window.
19. Click the **Import** button at the bottom of the window to save the code and close the window.
20. Ensure that the following values appear for the PATCH operation response:

Field	Value
Body	*BusinessData.ResponseTransStatusUpdate*
Media Type	*application/JSON*

21. Click **Apply.**
22. Click **Save.**

Step 5: Add the GET Operation for Transaction Base Data

Note: Before starting this procedure, be sure to complete the procedure “Adding a PATCH Operation for Transaction Statuses.”

The *getTransactionBaseData* operation gets general transaction data that is found in all transactions, such as the permit type, the permit status, and the permit applicant for a permit transaction.

1. Expand the **TransactionResource** resource.
2. In the **Operations** section, click the **Add** button and then select **GET operation** from the drop-down menu.
3. Click the new **GET** operation.
4. Enter the following information:

Field	Value
Name	*getTransactionBaseData*
Path	*{transResource}/{transRecordKey}*

Field	Value
	Although you can choose different names for the resource name and record key parameters, this procedure assumes that you use the given values.
Description	*Get base transaction data, such as applicant information*

5. Click **Request**
6. In each row of the **Parameters** list, click the *Enter a description* text and enter a description.
   These are example descriptions:

Parameter	Description
transResource	*Transaction Resource Name*
transRecordKey	*Transaction Record Key*

7. Click **Response.**
8. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
9. Enter *ResponseTransBase* in the **Name** field.
10. Click **Schema.**
11.  Click the **Import from File** icon next to the **Schema** button.
12. Locate and upload the *ResponseTransBase.json* file that you downloaded from My Oracle Support.
    The imported JSON code appears in the Import Business Object from JSON window.
13. Click the **Import** button at the bottom of the window to save the code and close the window.
14. Ensure that the following values appear for the GET operation response:

Page Element	Value to Enter
Body	*BusinessData.ResponseTransBase*
Media Type	*application/JSON*

15. Click **Apply.**

Step 6: Add the GET Operation for Transaction Fields Data

The *getTransactionFieldsData* gets field data from the application intake form configured using the Intake Form Designer.

1. In the **Operations** section of the Transactions Resource, click the **Add** button and then select **GET operation** from the drop-down menu.
2. Click the new **GET** operation.

The new GET operation has the default name of *GetTransactionResources.*

3. Enter the following information:

Field	Value
Name	*getTransactionFieldsData*
Path	*{transResource}/{transRecordKey}/child/FieldGroups* Although you can choose different names for the resource name and record key parameters, this procedure assumes that you use the given values.
Description	*Get specific transaction data, such as job cost.*

4. Click **Request.**
5. In each row of the **Parameters** list, click the *Enter a description* text and enter a description.

   These are example descriptions:

Parameter	Description
transResource	*Transaction Resource Name*
transRecordKey	*Transaction Record Key*

6. Click **Response.**
7. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
8. Enter *ResponseTransFields* in the **Name** field.
9. Click **Schema.**
10.  Click the **Import from File** icon next to the **Schema** button.
11. Locate and upload the *ResponseTransFields.json* file that you downloaded from My Oracle Support.

    The imported JSON code appears in the Import Business Object from JSON window.
12. Click the **Import** button at the bottom of the window to save the code and close the window.
13. Ensure that the following values appear for the GET operation response:

Page Element	Value to Enter
Body	*BusinessData.ResponseTransFields*
Media Type	*application/JSON*

14. Click **Apply.**
15. Click **Save.**

Step 7: Add the GET Operation for Transaction Data

The *getTransactionData* combines getTransactionBaseData and getTransactionFieldsData into a single operation, which you can use instead of using getTransactionBaseData and getTransactionFieldsData separately.

1. In the **Operations** section, click the **Add** button and then select **GET operation** from the drop-down menu.
2. Click the new **GET** operation.

   The new GET operation has the default name of *GetTransactionResources.*
3. Enter the following information:

Field	Value
Name	*getTransactionData*
Path	*{transResource}/{transRecordKey}* Although you can choose different names for the resource name and record key parameters, this procedure assumes that you use the given values.
Description	*Combines getTransactionBaseData and getTransactionFieldsData into a single operation.*

4. Click **Request.**
5. In each row of the **Parameters** list, click the *Enter a description* text and enter a description.

   These are example descriptions:

Parameter	Description
transResource	*Transaction Resource Name*
transRecordKey	*Transaction Record Key*

6. Click **Response.**
7. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
8. Enter *ResponseTransactionData* in the **Name** field.
9. Click **Schema.**
10.  Click the **Import from File** icon next to the **Schema** button.
11. Locate and upload the *ResponseTransactionData.json* file that you downloaded from My Oracle Support.

    The imported JSON code appears in the Import Business Object from JSON window.
12. Click the **Import** button at the bottom of the window to save the code and close the window.

13. Ensure that the following values appear for the GET operation response:

Page Element	Value to Enter
Body	*BusinessData.ResponseTransactionData*
Media Type	*application/JSON*

14. Click **Apply.**
15. Click **Save.**

Step 8: Add the GET Operation for Transaction Assignee

The *getTransactionAssignee* operation gets the assigned planner for transactions within the Planning and Zoning offering. You can configure subsequent tasks in the workflow process to reference the retrieved and stored getTransactionAssignee value.

1. In the **Operations** section, click the **Add** button and then select **GET operation** from the drop-down menu.
2. Click the new **GET** operation.

   The new GET operation has the default name of *GetTransactionResources.*
3. Enter the following information:

Field	Value
Name	*getTransactionAssignee*
Path	*publicSectorTransactionLatestAssignees/{transRecordKey}* Although you can choose different names for the resource name and record key parameters, this procedure assumes that you use the given values.
Description	*Get specific transaction data, such as job cost.*

4. Click **Request.**
5. In each row of the **Parameters** list, click the *Enter a description* text and enter a description.

   These are example descriptions:

Parameter	Description
transRecordKey	*Transaction Record Key*

6. Click **Response.**
7. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.

8. Enter *ResponseTransactionFields* in the **Name** field.
9. Click **Schema.**
10.  Click the **Import from File** icon next to the **Schema** button.
11. Locate and upload the *ResponseTransAssignee.json* file that you downloaded from My Oracle Support.
    The imported JSON code appears in the Import Business Object from JSON window.
12. Click the **Import** button at the bottom of the window to save the code and close the window.
13. Ensure that the following values appear for the GET operation response:

Page Element	Value to Enter
Body	*BusinessData.ResponseTransAssignee*
Media Type	*application/JSON*

14. Click **Apply.**
15. Click **Save.**

Step 9: Add the Transaction Type Resource

Note: Before starting this procedure, be sure to complete the previous procedures.

1. Access the main console in OIC.
2. In the list of OIC applications, click the application for your workflow process.
3. Click the **Integrations** option in the left frame.
4. Click the **TransactionsConnector** integration.
5. In the header of the Resources section, click **Add** to create a new transaction type resource.
6. Expand the new Resource section that appears, and enter the following information:

Field	Value
Name	TransactionTypeResource
Path	publicSectorRecordTypes

Step 10: Add the GET Operation for Transaction Type Data

The *GetTransactionTypeData* operation gets data that is associated with the transaction type definition rather than with an individual transaction. For example, this operation can get the overall fee structure for a permit definition, because the fee structure is associated with the permit type.

To set up the GET operations for transaction type data:

1. In the **Operations** section of the Transaction Type Resource, click the **Add** button and then select **GET operation** from the drop-down menu.

2. Click the new **GET** operation.
3. Enter the following information:

Page Element	Value
Name	*getTransactionTypeData*
Path	*{transResource}*
Documentation	*Get transaction type setup data.*

4. Click **Request**
5. In the **Parameters** list, click the *Enter a description* text and enter a description.

   Here is an example description:

Parameter	Description
transResource	*Transaction Resource Name*

6. Click **Response.**
7. Click the **+** icon next to the **Body** field to open the Import Business Object from JSON window.
8. Enter *ResponseTransTypeData* in the **Name** field.
9. Click **Schema.**
10.  Click the **Import from File** icon next to the **Schema** button.
11. Locate and upload the *ResponseTransType.json* file that you downloaded from My Oracle Support.

    The imported JSON code appears in the Import Business Object from JSON window.
12. Click the **Import** button at the bottom of the window to save the code and close the window.
13. Ensure that the following values appear for the GET operation response:

Page Element	Value to Enter
Body	*BusinessData.ResponseTransTypeData*
Media Type	*application/JSON*

14. Click **Apply.**
15. Click **Save.**

Setting Up Process Definitions for Workflow

Workflow manages status updates throughout the transaction lifecycle and is an essential part of your setup. This topic provides information for creating your workflow process definitions.

Note: The procedures in this topic relate to the specific requirements of Public Sector Compliance and Regulations workflow. To create your workflow, you first need to become familiar with OIC and, in particular, the process builder in OIC. For more information, refer to your OIC documentation at *https://docs.oracle.com/en/cloud/paas/integration-cloud/user-processes/develop-structured-processes.html*

Note: Currently, in the context of data object parameters, data association parameters, and REST resource attributes, the terms *record, transaction,* and *permit* are interchangeable. The abbreviation *trans* is often used to represent *transaction.*

Setting Up Data Objects for a Process

Data objects provide a structure for storing data sent from the Public Sector Compliance and Regulation system. Every process definition that you create needs the same data objects, including:

- Simple string data definitions to store identifying information about the transaction and transaction type.
- Business object data definitions to store transaction base data, field data, and transaction type data.
  The data definition for fields includes *all possible* fields that can be added to an intake form, even though the transaction may use only a subset of fields that are appropriate for the type of transaction. Any fields that are not part of a specific transaction type remain blank when the workflow process retrieves the field data.

You must set up your data objects before you continue to this topic's additional procedures for defining data associations.

Note: Before you set up your data objects, you need to set up the transaction connector for the application. This is because the transaction connector's GET operations provide the underlying schema for the data. Setting up the transaction connector is described in the topic *Setting Up the Transactions Connector*.

To set up data objects for a process definition:

1. Access the process definition in OIC.
2. Click **Data Objects.**
3. Set up the data definition for transaction base data:
   a. In the Data Objects window, click **Add.**
   b. In the Create Process Data Object window, enter the following information:

Page Element	Value
Name	*transactionBaseData*

Page Element	Value
Data Type	*Business*
The drop-down list for data types	*BusinessData.ResponseTransBase*

c. Click **Create** to create the data definition and return to the Data Objects window.

4. Set up the data definition for transaction field data:

a. Click **Add.**
b. Enter the following information:

Page Element	Value
Name	*transactionFieldsData*
Data Type	*Business*
The drop-down list for data types	*BusinessData.ResponseTransFields*

c. Click **Create.**

5. Set up the data definition for transaction type data:

a. Click **Add.**
b. Enter the following information:

Page Element	Value
Name	*transactionTypeData*
Data Type	*Business*
The drop-down list for data types	*BusinessData.ResponseTransType*

c. Click **Create.**

6. Click **Close** to close the Data Objects window.
7. Click **Save.**

Creating a Data Object to Store Start Event Arguments

In the next task you will define the start arguments for the Start event. In this procedure, you create the structure to store the arguments.

This procedure involves:

- Creating a business type using the business object option.
- Creating a data object and associating it with the business object.

To create the data object for start event arguments:

1. Open the process definition.
2. Select Business Types from the left panel.

   > **Note:** Notice the other business types created automatically when importing the downloaded JSON, such as ResponseTransactionData.

3. Click **Edit** in the header, which causes the **Create** button to appear.
4. Click **Create,** and select *New Business Object.*
5. On the Create Business Object dialog box, enter the business object name, such as *InitTransaction,* and select the Parent Module as *BusinessData.*
6. Click **Next.**
7. Use the **Add Attribute** button to add these string data types to the business object.

Attribute Name	Data Type
transactionKey	*String*
transactionType	*String*
externalBaseURL	*String*
resourceName	*String*
transactionOwner	*String*
transactionId	*String*
classification	*String*
subclassification	*String*

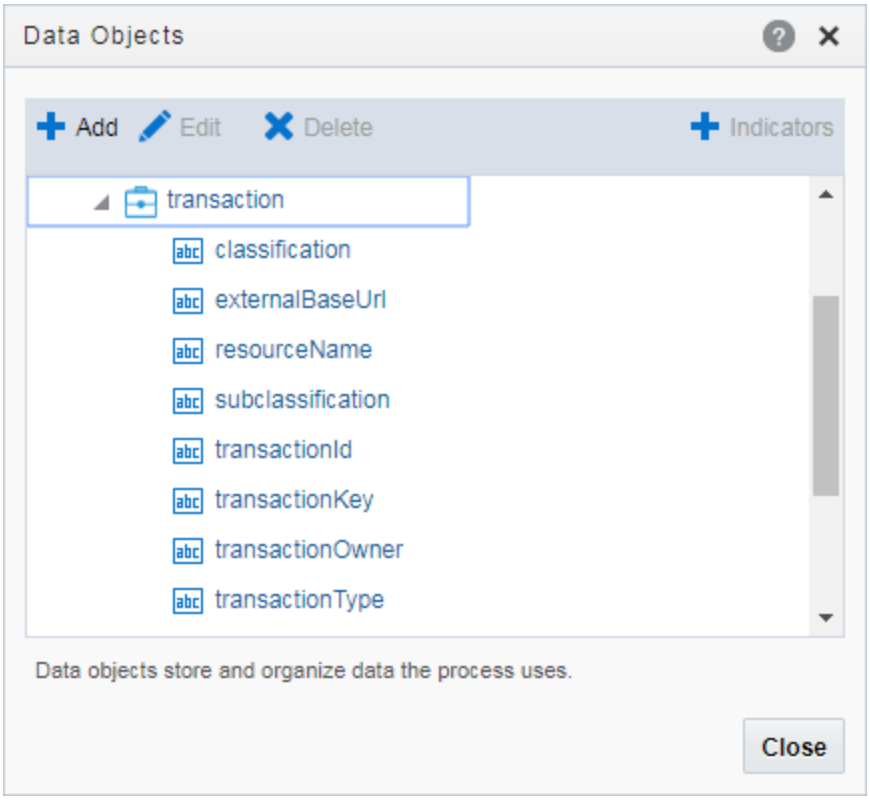
8. Click **Finish.**
9. Return to your process definition diagram.

10. Click the **Data Objects** button.
11. Click **Add.**
12. On the Create Process Data Object dialog box, make these changes:

Field	Value
Name	*transaction*
Data Type	*Business*
drop-down list	*BusinessData.InitTransaction*

13. Click **Create.**

This example illustrates the expanded transaction business data object, with the required string data within it.



Defining Arguments for the Start Event

When a transaction, such as a permit intake application, is submitted, the software instantiates that transaction's workflow process by passing parameters, such as the transaction ID, to OIC. The Start event in your process diagram must have arguments defined for these parameters.

To set up the arguments for the start event:

1. Select the Start event in the process definition.

2. Click the Start event, and select *Open Properties.*

   The default view is the General section of the Implementation Properties.
3. In the **How do you want to implement it?** section, select *Define Interface* as the **Type.**
4. Click the pencil icon next to the **Type** field to open the Configure window.
5. Enter an operation name, such as *start.*
6. Add the following rows to the **Arguments Definition.**

   a. Use the **Add** button to add these strings using the values in this table, where each row represents a separate argument you need to create:

Name	Type
TransactionKey	*string*
TransactionType	*string*
ExternalBaseURL	*string*
ResourceName	*string*
TransactionOwner	*string*
TransactionId	*string*
Classification	*string*
Subclassification	*string*

   > **Note:** Arguments added to the Start event must be named *exactly* as documented.

   > **Note:** You need to complete the output data association if you want the argument values stored in a Data Object.

   b. Click **OK.**
7. Close the properties panel and click **Save.**

Defining Data Associations for the Start Event

The data associations for the Start event capture identifying information about the transaction for initiating the process instance.

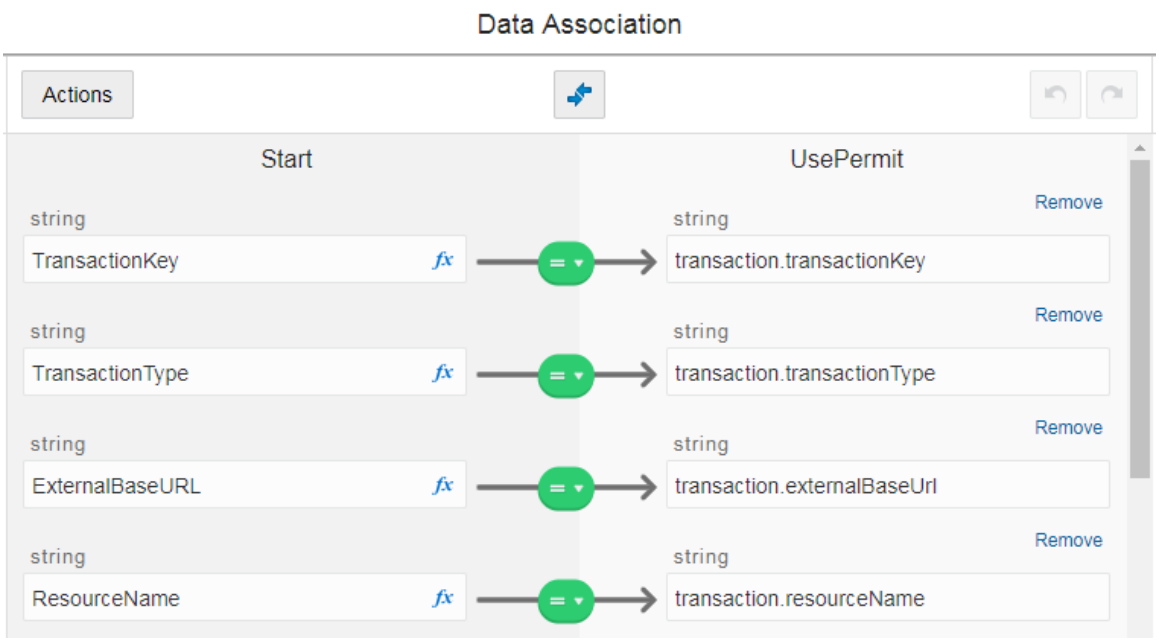
In this task, you map the arguments for the Start event to the data object values you entered for the *transaction* object.

To set up the data associations:

1. Open the process definition and select the Start event.
2. Click the **Data Association** button.
3. Set up the following data associations, mapping the Start event arguments to the appropriate attributes in transaction business object.

Source Data (Left side of the map)	**Target Data (Right side of the map)**	**Description**
TransactionKey	*transaction.transactionKey*	The transaction ID.
TransactionType	*transaction.transactionType*	The transaction type ID.
ExternalBaseURL	*transaction.externalBaseURL*	The URL for the Public Sector Compliance and Regulation system.
ResourceName	*transaction.resourceName*	The name of the REST API resource.
TransactionOwner	*transaction.transactionOwner*	The owner of the transaction.
TransactionId	*transaction.transactionId*	The transaction ID.
Classification	*transaction.classification*	The classification of the transaction.
Subclassification	*transaction.subClassification*	The subclassification of the transaction.

This example illustrates mapping the Start event arguments to the associated data attributes in the transaction business object.



4. Click **Apply.**
5. Click **Save.**

Defining Data Associations for Sending Notifications

The data associations for a notification task define the information that the task sends to the public sector communications center.

> **Note:** Create your email templates in the communications center before you set up integration for notification workflow tasks.

For more information about the communications center, see *Setting Up Communication Events*.

To set up the data associations:

1. Access the process definition and select the system task.
2. Click the **Data Association** button.
3. Set up the following input data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
[event name] For example, *"LNP_WORKFLOW_001"*	*body.eventCode*	The event as defined in the Communications Center in the public sector system. The source data string must be in quotation marks, and it must exactly match the identifier of an event.

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
		Oracle delivers these communication events, *<Offering>_WORKFLOW_001* through *<Offering>_WORKFLOW_005.* Where "Offering" refers to your offering code, such as LNP for Permits or PNZ for Planning and Zoning.
[template name] For example, *"Application_Accepted"*	*body.templateCode*	The identifier for the template to be used for the email. The source data string must be in quotation marks, and it must exactly match the name of a template in the transaction application.
"LnpRecordKey"	*body.recordFirstKeyName*	The name of the key field.
transaction.transactionKey	*body.recordFirstKeyValue*	The transaction ID.
true or *false*	*body.email*	This Boolean field indicates whether the notification is sent as an email. Enter *true* only if the template is an email template.
true or *false*	*body.notification*	This Boolean field indicates whether the notification is sent as an in-product notification. . Enter *true* only if the template is an in-product notification template.

> **CAUTION:** Templates are associated with either email or in-system notifications. Be sure to set up the body.email and body.notification values properly. Exactly one of the values must be true. If you want to send both types of notifications, you need to create two notification tasks.

4. Click **Apply.**
5. Click **Save.**

Defining Data Associations for Sending Status Updates

The data associations for a status update task define the information that the task sends to the Public Sector Compliance and Regulation system.

To set up the data associations:

1. Access the process definition and select the system task that updates the transaction status.
2. Click the **Data Association** button.
3. Set up the following input data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
transaction.resourceName	*resourceName*	The unique system identifier for the transaction type.
transaction.transactionKey	*transactionKey*	The transaction ID
[new transaction status] For example: *"Accepted"*	*body.status*	The status to be assigned to the transaction. The source data string must be in quotation marks, and it must exactly match one of the valid statuses for the transaction application.

4. Click **Apply.**
5. Click **Save.**

Defining Data Associations for Retrieving Transaction Base Data

The data associations for a task that retrieves transaction base data provides a structure for storing the retrieved data. In this task you create both input and output data associations.

To set up the data associations:

1. Access the process definition, and select the system task that retrieves transaction base data.
2. Click the **Data Association** button.
3. Set up the following input data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
transaction.resourceName	*Resource*	The unique system identifier for the transaction type.
transaction.transactionKey	*TransactionKey*	The transaction ID

4. Click **Output.**
5. Set up the following output data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
body	*BaseData*	This business object contains all of thebase data. Mapping individual fields would be much more complex and is not necessary.

6. Click **Apply.**
7. Click **Save.**

Defining Data Associations for Retrieving Transaction Field Data

The data associations for a task that retrieves transaction field data provides a structure for storing the retrieved data. In this task you create both input and output data associations.

To set up the data associations:

1. Access the process definition, and select the system task that retrieves transaction field data.
2. Click the **Data Association** button.
3. Set up the following input data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
transaction.resourceName	*transactionResource*	The unique system identifier for the transaction type.
transaction.transactionKey	*TransactionKey*	The transaction ID.

4. Click **Output.**
5. Set up the following output data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
body	*transactionFieldsData*	This business object contains all of the transaction fields. This includes all fields that can be included on the application intake form, whether or not the field exists for a specific transaction. Individual fields are nested within the *items* object. You can't expand the *items* object on this page, but they are

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
		available in the expression editor that you use when creating business logic based on transaction data. Mapping individual fields would be much more complex and is not necessary.

6. Click **Apply.**
7. Click **Save.**

Defining Data Associations for Retrieving Transaction Type Data

The data associations for a task that retrieves transaction type data provides a structure for storing the retrieved data. In this task you create both input and output data associations.

To set up the data associations:

1. Access the process definition, and select the system task that retrieves transaction type data.
2. Click the **Data Association** button.
3. Set up the following input data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
transaction.resourceName	*transactionResource*	The unique system identifier for the transaction type.

4. Click **Output.**
5. Set up the following output data associations:

Source Data (Left side of the map)	Target Data (Right side of the map)	Description
body	*transactionTypeData*	This business object contains all transaction type fields.

6. Click **Apply.**
7. Click **Save.**

Defining Statuses (Outcomes) for Human Tasks

The **Action** property for a human task lists the possible outcomes of the task. The actions you define appear as values in the **Task Status** drop-down list box on the Workflow page where agency staff manages workflow tasks. When the

task status is updated on the Workflow page, OIC recognizes it as the task outcome and continues to the next step or gateway.

To define status values representing the outcomes of human tasks:

1. Access the process definition and select the human task.
2. Open the task properties.
3. In the **Action** field, enter a comma-delimited list of status values.
   Do not put a space before or after the comma. For example, if the status are *Accept, Reject,* and *More Information,* enter *Accept,Reject,More Information* in the **Action** field.
4. Close the properties window and save.

Defining Conditional Logic for Gateways

In a process map, gateways represent decision points where there is a branch in the process flow. The logic for taking different paths after the gateway is associated with the arrows to the possible subsequent tasks.

An arrow that represents a default branch does not require any logic.

For all other arrows, you need to set up the conditions under which the branch is used. To do this:

1. Access the process definition and select the arrow.
2. Click the pencil icon for the arrow to open the arrow properties.
3. Select the **Conditional Flow** check box.
   This check box is selected for all arrows other than the default arrow after a gateway.
4. Click the pencil icon for the **Condition** field.
5. Use the Expression Editor window to specify the conditions for using this branch.
   The Data Objects tab provides access to the data elements that you can evaluate. Transaction field data (the data from the intake form) is nested within the *items* element under *TransactionFieldsData.*
   In expressions that look for an exact match, take extra care with the spelling, capitalization, and punctuation of values that the expression evaluates.
6. Select the gateway and open the gateway properties.
7. Use the **Order** property to specify the order in which the previous task's outcomes are evaluated for purposes of determining which arrow to follow.

Configuring a Planning and Zoning Human Task to Reference the Assigned Planner Value

If you want to assign the user to a human task, you need to have first setup the Transaction Connector to use the getTransactionAssignee operation.

For more information on setting up the Transaction Connector, see *Setting Up the Transactions Connector*.

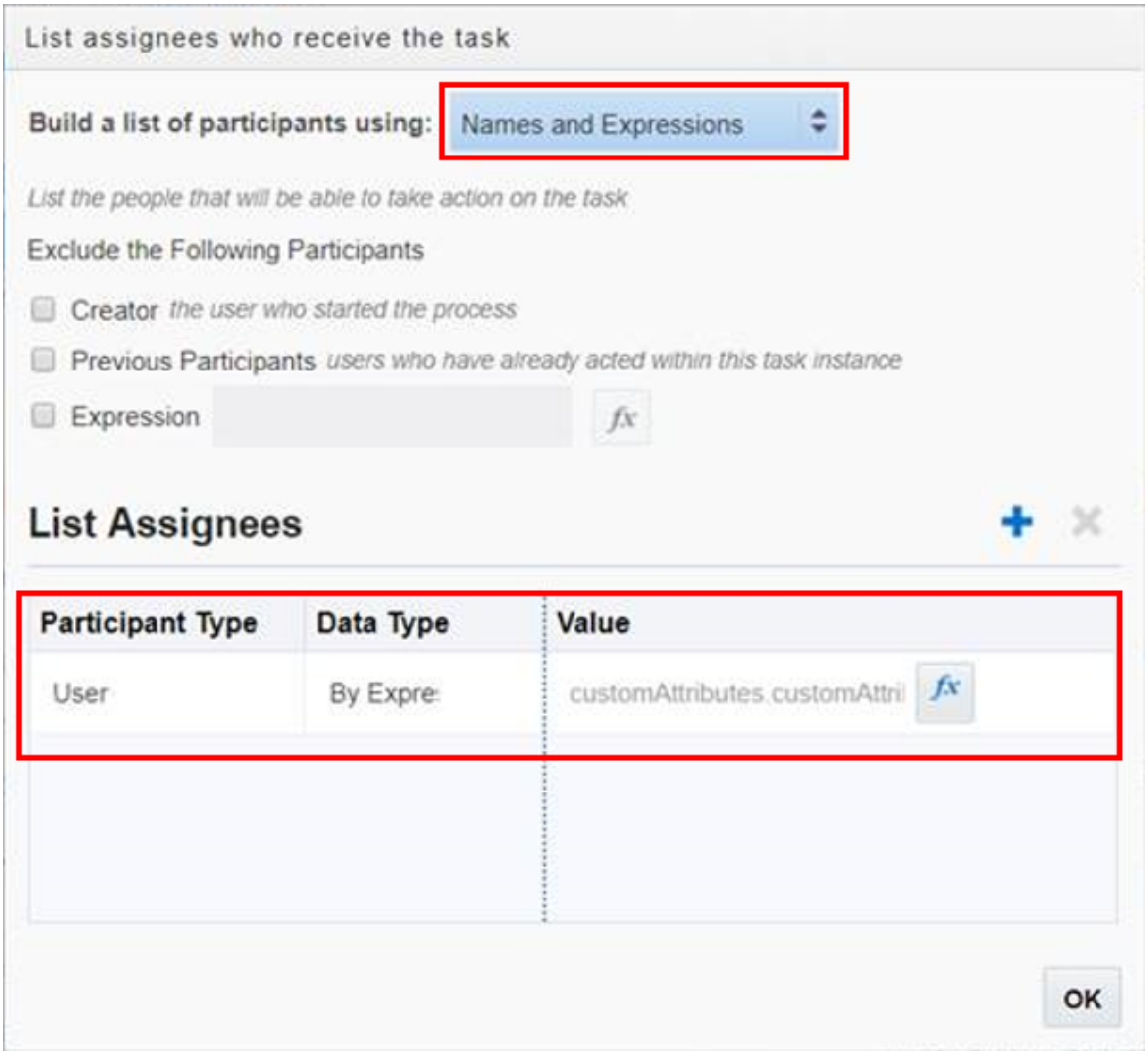
To associate the transaction assignee data to your workflow process:

1. When configuring the workflow process data object, create a data object to store the output of the service call to getTransactionAssignee.
2. Locate the first human task in the process after the service call (Start event), and open the Data Association interface.
3. On the Input Data Association, map the assignee data object to execData.customAttributes.customAttributeString1.

4. For subsequent human task to which you want to reference the assigned planner, set the Assignee(s) value in the General Properties to *customAttributes.customAttributeString1.*

    a. Open the properties for the human task.
    b. Select Implementation > General.
    c. Click the Edit icon for the Assignee(s) field.
    d. Select *Names and Expressions* in the **Build a list of participants using** list.
    e. Click Add for the List Assignees edit box.,
    f. Select *Add User.*
    g. Click in the Data Type column, and select *By Expression*.
    h. Click the **fx** button in the Value column, and select *customAttributes.customAttributeString1.*
    i. Click **OK.**

This example illustrates the settings on the List assignees who receive the task window.



5. Click **Save.**

Implementing the Expiration Feature

If you are implementing the expiration feature for Permits or Planning and Zoning, you need to add these system tasks to the workflow process definition in Oracle Integration Cloud:

System Task	Description
Accepted	Updates the transaction status to *Accepted.* This task should be placed early in the process flow.
Completed	Updates the transaction status to *Completed* or *Certified.* This task should be placed at the end of the process flow.

Note: If you are updating an existing process definition to incorporate these system tasks, make sure to activate the new version using a new version number, and make sure to reference that version of the process definition from your transaction type definitions.

Using Custom Properties

When you set up workflow, a variety of custom properties are available for implementing various features. This topic describes how use the custom properties.

These custom properties are available for workflow and are described in more detail below:

Property	Usage	Values
PSC_LIST_ORDER	Use this property to set the order for human tasks when there are multiple possible paths through the process definition. The order does not affect the workflow process, but it allows users to see the possible future workflow tasks in a logical order.	Integers For example: `PSC_LIST_ORDER: 2`
PSC_FINAL_ACTIVITY	Use this property to identify a human task that is not allowed to progress when the transaction has a condition that applies the **Prevent Issue or Final** business rule. In particular, use this property to identify the final human task in the process definition. See *Setting Up Conditions* and *Applying Conditions to Applications*.	*Yes* identifies the final activity. A blank value or a *No* value means that the task is not the final activity. For example: `PSC_FINAL_ACTIVITY: Yes`

Property	Usage	Values
PSC_ACTIVITY_TYPE	Use this property to identify the final inspection task in the process. Setting this property is necessary to support the transaction business logic that auto-advances the inspection task when the last inspection is closed.	*Inspection* is the only value with related business logic. Leave this property blank for other types of activities.
PSC_AUTO_UPDATE_ACTION	Use this property to identify the action to take when auto-advancing the final inspection task in a process. Setting this property is necessary to support the transaction business logic that auto-advances the inspection task when the last inspection is closed. **Note:** Security roles related to the Mobile Inspector app and the Plan Reviewer should be included in the task swimlane mapping.	The exact action name as specified in the Action property for the human task. Take extra care with the spelling, capitalization, and punctuation of the action name. For example: `PSC_AUTO_UPDATE_ACTION: Approve`
PSC_UNRESTRICTED_ACTIONS	Identifies actions that can be taken for the task even if there is logic preventing the task from advancing. Use this property when the task includes possible outcomes that return to a previous task rather than advancing. For example, if the final inspection can be advanced with an action of “Approve” or returned with an action of “Needs More Information,” then use the PSC_UNRESTRICTED_ACTIONS property to make “Needs More Information” an unrestricted action. This allows users to take the “Needs More Information” action, even though the logic that prevents the final inspection task from advancing disallows other actions. Currently, the PSC_UNRESTRICTED_ACTIONS property is applicable only for the Inspection action, as identified by the PSC_ACTIVITY_TYPE.	When there are multiple unrestricted actions, separate the actions with commas but no spaces. This is the same format used in the Actions property where you define all of the available actions for a human task. For example: `PSC_UNRESTRICTED_ACTIONS: Needs more Info, Proceed, OK`

Making Custom Properties Available in a Process Definition

Before you can use a custom property, you need to add the property to the process definition. You must do this for each of your process definitions.

To add a custom property to a process definition:

1. Access the process definition and click the **#** (Custom Properties) toolbar icon.
2. Enter the following values in the **Property Name** and **Description** fields in the Custom Properties list:

> **Note:** You must use the exact property names given in this procedure. You can, however, alter the descriptions.

Property Name	Description
PSC_LIST_ORDER	Human tasks display order
PSC_FINAL_ACTIVITY	Identify the final human task
PSC_ACTIVITY_TYPE	Identify the type of activity such as inspection
PSC_AUTO_UPDATE_ACTION	Identify the action to take when auto-advancing

3. Click **OK.**
4. Click **Save.**

PSC_LIST_ORDER Property

The Workflow page for a permit includes an option to view a list of all past, present, and not started human tasks for the permit. The list displays past and present tasks in chronological order. However, the chronology for tasks that haven't been started is not necessarily fixed. The branching logic in a process means that some tasks might be omitted or might occur in a different order depending on permit data or on the outcome of previous tasks.

To control the order in which not started human tasks appear, use the PSC_LIST_ORDER custom property. On the Workflow page's list view, tasks that have not yet started are listed in the order you specify. If multiple not started tasks have the same number, they appear in the list in random order.

To assign order numbers to human tasks:

1. Analyze all human tasks in the process and decide on the appropriate order.

   Tasks appear in ascending numerical order. You assign order numbers one task at a time, so if you later decide to change the order, you have to update each affected task individually.
2. Access the process definition.
3. Select a human task.
4. Open the task properties.
5. Select the **Business Properties** icon in the icon bar, then select **Custom Properties** from the list of options that are associated with that icon.
6. Enter a number in the **PSC_LIST_ORDER** custom property.
7. Close the Properties window and save.
8. Repeat the previous steps for all human tasks in the process definition.
9. Click **Save.**

PSC_FINAL_ACTIVITY Property

Human tasks that you identify as a final activity cannot advance when a permit has a condition that applies the **Prevent Issue or Final** business rule.

To identify the final human task using the PSC_FINAL_ACTIVITY property:

1. Access the process definition.
2. Select the final human task.
3. Open the task properties.
4. Select the **Business Properties** icon in the icon bar, then select **Custom Properties** from the list of options that are associated with that icon.
5. Enter *Yes* in the **PSC_FINAL_ACTIVITY** custom property.
   To remove a *Yes* value from this property, set the value to *No* or clear the value and leave it blank.
6. Close the Properties window and save.
7. Click **Save.**

PSC_ACTIVITY_TYPE and PSC_AUTO_UPDATE_ACTION Properties

Permit processing includes logic to automatically progress past the final inspection step in the process definition when permit inspections are complete. To enable this functionality, you must identify the final inspection task in the process definition using the PSC_ACTIVITY_TYPE custom property. Further, you must identify the workflow action to apply to that task using the PSC_AUTO_UPDATE_ACTION custom property.

To set up custom properties for auto-advancing an inspection task:

1. Access the process definition.
2. Select the human task that represents final inspections.
3. Open the task properties.
4. Select the **Business Properties** icon in the icon bar, then select **Custom Properties** from the list of options that are associated with that icon.
5. Enter *Inspection* in the **PSC_ACTIVITY_TYPE** custom property.
6. Enter the desired action in the **PSC_AUTO_UPDATE_ACTION** custom property.
   The action you enter is the action to be taken when the task is successfully complete—that is, when the permit passes its final inspection. Take care to use the correct spelling, capitalization, and punctuation for the action name.
7. Close the Properties window and save.
8. Click **Save.**

Mapping Workflow Swimlanes to Roles

This topic describes how to assign security roles to the swimlanes in your workflow process definition.

In workflow process definitions, swimlanes represent roles. After the OIC application containing the process definition is activated, use the Manage Roles functionality in OIC to map the swimlanes to roles. The mapping applies to all process definitions in the OIC application.

A swimlane is typically associated with security roles, and it can be associated with multiple roles if needed. It can also be associated with one or more individual users if that approach is more applicable. A swimlane determines who is responsible for carrying out a task.

For example, if a swimlane is for plan review, you would add the PSC Plan Reviewer role to that swimlane. Likewise, if another swimlane is for inspecting, you would add all the roles that apply to that swimlane, such as PSC Building Inspector and any roles related to the Mobile Inspector app.

Note: When supervisors assign or reassign tasks, they can only assign the task to agency staff associated with security roles that are assigned to the swimlane in the underlying workflow process definition. The swimlane that contains the Start event node needs to be mapped to the PSCR Submitter Group. The procedure below uses that scenario as an example to illustrate the process used to map a security role to a swimlane.

To map swimlanes to roles:

1. Access the My Tasks area of Oracle Autonomous Integration Cloud.
2. Click My Tasks in the left navigation menu, and click Workspace in the My Tasks header.
3. Click Administration in the left navigation menu.
4. If the Manage Roles page does not appear by default, click **Manage Roles** in the left navigation menu.

   The Manage Roles page lists process roles using the format *[application].[swimlane].*
5. Search for your application to filter the list.

   The **Process Owner** and **Process Reviewer** roles are part of all applications. Other swimlanes in the list are ones that you created in your process definitions.
6. Add the delivered role *PSCR Submitter Group* to the swimlane that contains the Start event for your process model:

   a. Select the swimlane that contains the **Start** task in your process definitions.

      In the delivered Solution Packages that Oracle provides, this swimlane is labeled *Applicant.* This swimlane applies to the user submitting a transaction, such as a permit application.
   b. In the **Assign Roles** list for the selected swimlane, click **Add Member**.
   c. In the dialog box for adding members, search by *Groups* for *PSCR Submitter Group.*

      A group in OIC is equivalent to a role in the Public Sector Compliance and Regulation system.
   d. In the search results, select *PSCR Submitter Group* and then click **OK** to assign the role to the swimlane and return to the list of swimlanes.
7. Click Save.

Monitoring Workflow Transactions

This topic describes how to use the Inconsistent Instances page to resolve discrepancies between the Public Sector Compliance and Regulation system and the Oracle Integration Cloud instance (OIC).

Working with Workflow Transaction Logs

During normal transaction processing, it's possible that either the Oracle Integration Cloud instance or the Oracle Public Sector instance can become momentarily unresponsive. While a rare occurrence, you need to be able to resolve any transactions that occurred during the down time to make sure the OIC instance and your Public Sector offering are synchronized.

For example, if OIC is unresponsive for a five minute period according to your system logs and notification system, you can use the Inconsistent Instances page to isolate any transactions that occurred specifically during that time to make sure individual transaction statuses are in sync with the associated workflow process.

Using the Inconsistent Instances Page

Use the Inconsistent Details page to search for transactions created during a specific time range.

Access the Inconsistent Instances page by selecting **Workflow and Transaction Log** > **Inconsistent Instances** .

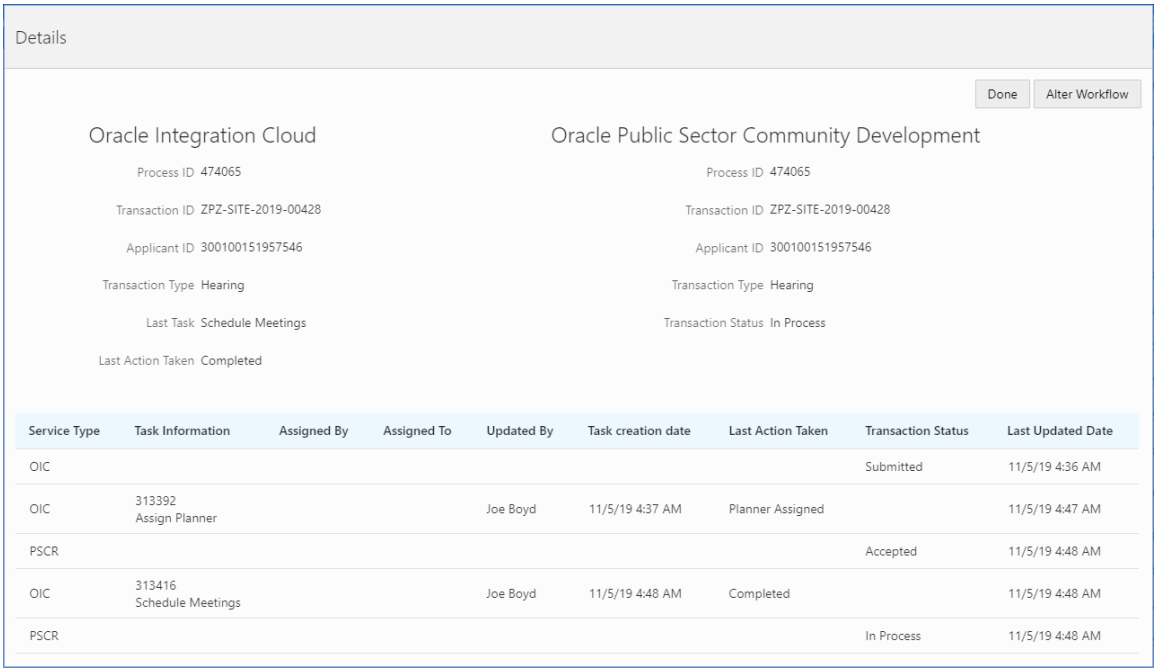
Page Element	Description
Which system is restored	Select the system that was not available for a period of time and needed to be restored from a back up. Options are: • *PSCR:* select if the Public Sector Compliance and Regulation system was down. • *OIC:* select if the Oracle Integration Cloud instance was down. If one of the systems is unavailable for a given period and needed to be restored, then the other system becomes the most current source of transaction status for any manual synchronization.
From/To	Use the date/time controls to specify a time range for isolating the affected transactions.
Workflow Instance ID	The workflow instance ID assigned for a specific transaction by the OIC system when that transaction is submitted.
Process Status	The current status of the process, such as *CANCELLED, OPEN, COMPLETED,* and so on.
Transaction	The unique ID of a specific submitted transaction, which is comprised of the topic type and the auto number rule.
Transaction Status	The current status of the individual transaction according to the corresponding workflow process definition, such as *Submitted, Inspection, Plan Review,* and so on.
Process Definition	The workflow process definition in OIC to which the specific transaction is associated. This value is comprised of the OIC instance name, the space name, and the workflow process application name as defined in OIC.
Transaction Update Date	The last time the transaction was updated.
Process Update Date	The last time the process was updated.

Using the Details Page

Use the Details page to drill into a specific transaction and take action to resolve any discrepancies between the OIC and PSCR systems. The Details page contains the transaction history in sequential order between the two systems for a given transaction.

Access the Details page by clicking on any row in the grid on the Inconsistent Instances page.

This example illustrates sample data as it would appear in the Details page.



At the top of the Details page the Oracle Integration Cloud and the Oracle Public Sector Compliance and Regulation sections display a high-level view of the current status for that transaction in the respective systems.

Depending on the information displayed, you may elect to perform different actions, which appear as buttons on the right hand side of the Details page.

Page Element	
Done	Closes the Details page with no action taken. Equivalent to Cancel.
Cancel Workflow	Cancels the transaction association with the underlying workflow so the transaction can be resubmitted. All previous work would be lost.
Alter Workflow	Enables the system administrator to alter the workflow manually to ensure both systems are in sync in the event of an momentary unavailability of either system. For more information on the options when altering workflow, see *Using Workflow*.
Resubmit Application	Enables the system administrator to resubmit an application.

Page Element	Description
Process ID	The workflow instance ID assigned for a specific transaction by the OIC system when that transaction is submitted.
Transaction ID	The unique ID of a specific submitted transaction, which is based on the auto number rule.
Applicant ID	The unique ID of the applicant who submitted the transaction intake form.

Page Element	Description
Transaction Type	The transaction type as defined on the Transaction Type page. For example, for a permit this will be the Permit Type value.
Last Task	The last task in the workflow process definition that the transaction has reached. (Applies only to the OIC system.)
Last Action Taken	The action taken on the last task. (Applies only to the OIC system.)
Transaction Status	The current status of the transaction reflected in the PSCR system, such as In *Process, Accepted,* and so on.

Page Element	Description
Service Type	Indicates what service handled the request. Options are: • *PSCR:* indicates the Public Sector system. • *OIC:* indicates the Oracle Integration Cloud instance.
Task Information	The task information associated with the task defined in the workflow process definition.
Assigned By	The user name of the individual to who assigned the current task to the assignee.
Assigned To	The user name of the individual to whom the current task is assigned.
Updated By	The user name of the individual who updated the task.
Task Creation Date	The date and time the task in the workflow process definition was initiated for the current transaction.
Last Action Taken	The last action taken as reflected in the OIC system.
Transaction Status	The status of the transaction reflected in the PSCR system.
Last Updated Date	The date and time when the transaction was last updated.

2 Configuring Fee Decision Models

Creating Decision Models for Fees

This topic describes the requirement of creating a decision model after creating fee items and before creating a fee schedule. You use Oracle Autonomous Integration Cloud to create decision models.

Prerequisites

Before you create a decision model, you need to create any required fee items that will be associated with the decision model.

For more information on fee items, see *Setting Up Fee Items*.

Decision Model Overview

You create decision models using the Oracle Autonomous Integration Cloud (OIC) decision modeling feature. Use this feature to create decision models to automate the decision logic in your business processes. As part of creating a decision model, add and order decisions, define decision inputs, and model the logic. The decision model editor supports the Decision Modeling and Notation (DMN) standard for you to create your models.

For more information on decision models refer to your Oracle Autonomous Integration Cloud documentation, *Create Decisions*.

In the Public Sector Compliance and Regulation services, a decision model enables you to automate the calculation of fees based on your business process criteria.

For example, assume your agency applies varying fees based on the total cost of a building project for which a permit is being requested. A decision model enables you to automate this business logic:

- If the project value is less than or equal to $500, then the application fee is $50.
- If the project value is more than $500 but less than or equal to $1,000, then the application fee is $75.
- If the project value is more than $1,000 and $5,000, then the application fee is $125.
- For any project value over $5,000, then the application fee is $200.

Before you create a decision model, you must first create a fee item. After creating the decision models, you can then associate the decision model with a fee schedule. The fees workflow generally follows these main steps and events:

1. Create fee item(s).
2. Create decision model based on existing fee item(s).
3. Create a fee schedule incorporating fee items and decision model.
4. Associate a fee schedule with a transaction type.
5. Map intake form fields to decision model in the Intake Form Designer.
6. When an end user is submitting an intake form the system applies fees and fee logic based on input.

Configuring Decision Models

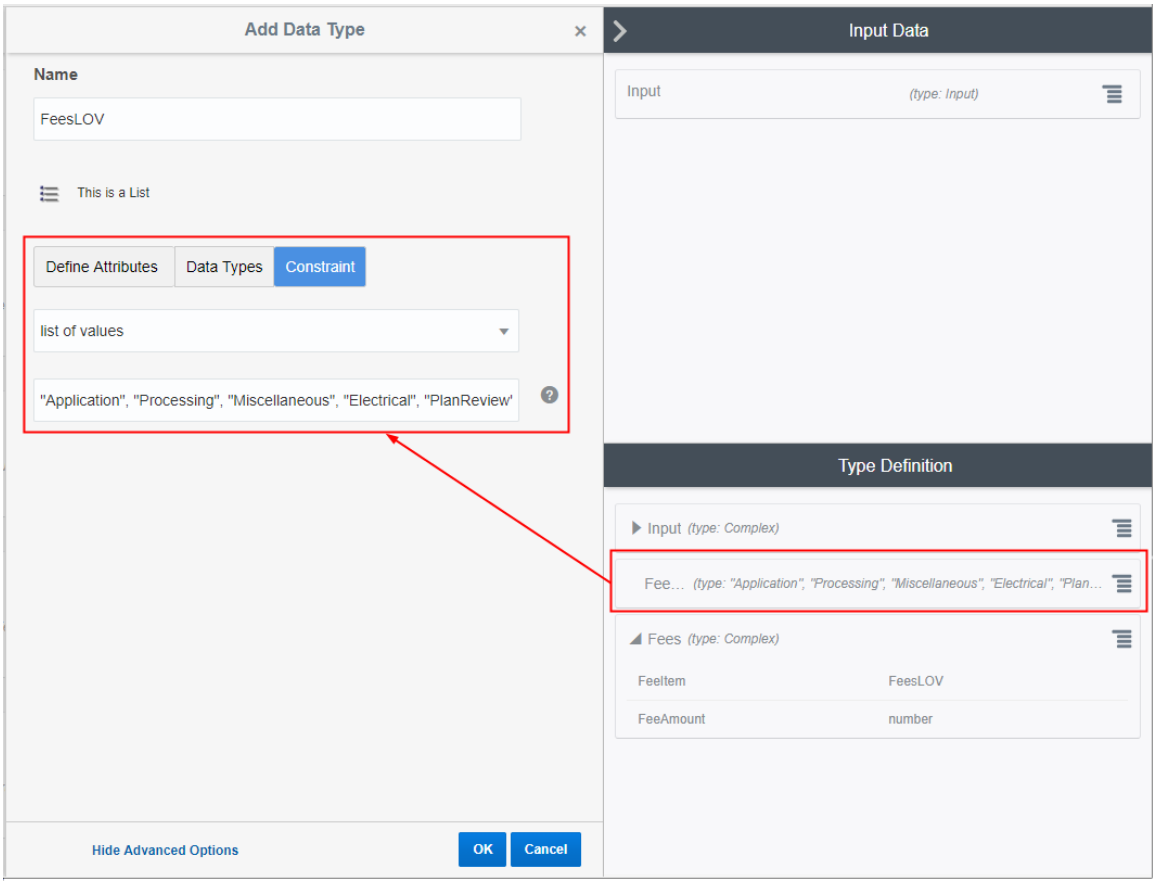
You can set up inputs and decisions any way you like following the guidelines provided in the documentation for OIC. However, the configuration here for output data types is required for setting up the fee schedule.

To configure decision models:

1. Create a fees list of values (LOV) with the fee item names.

   Currently, there is no direct integration of configuration data, such as fee items, between OIC and Oracle Public Sector Compliance and Regulation. Although creating an LOV is optional, any fee item names added to a decision model output need to be entered exactly as they appear in Oracle Public Sector Compliance and Regulation. When you use the LOV and enter values in a decision output. OIC validates the entry and displays a warning if your entry does not match an item in the LOV.

   This example illustrates the list of values setup in OIC used to validate fee items entered in the **Decision Table**. Select *list of values* in the **Constraint** options when you're adding the data type definition.



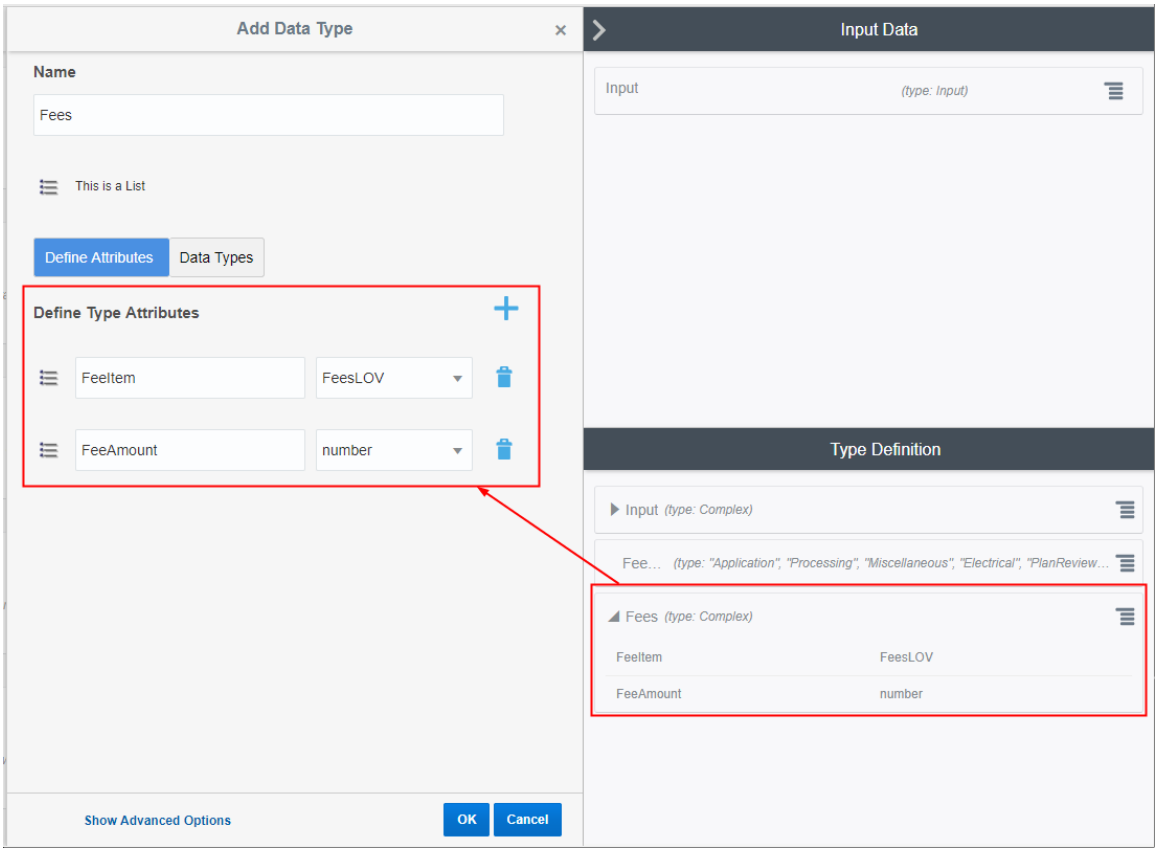
2. (Required) Set up the complex data type output to include these attributes: *FeeItem* and *FeeAmount*.

   The *FeeItem* attribute should use the fees LOV that you created. The *FeeAmount* attribute uses numbers that you enter on the decision model.

   > **Note:** This step is important because the *FeeItem* and *FeeAmount* attributes are used to map the fee items on the fee schedule to the decision model.

   > **Note:** Incorporating values derived from fields defined as reusable in your intake form is not supported.

   This example illustrates the complex data type output that you set up in OIC for your decision model. You must define *FeeItem* and *FeeAmount* attributes.

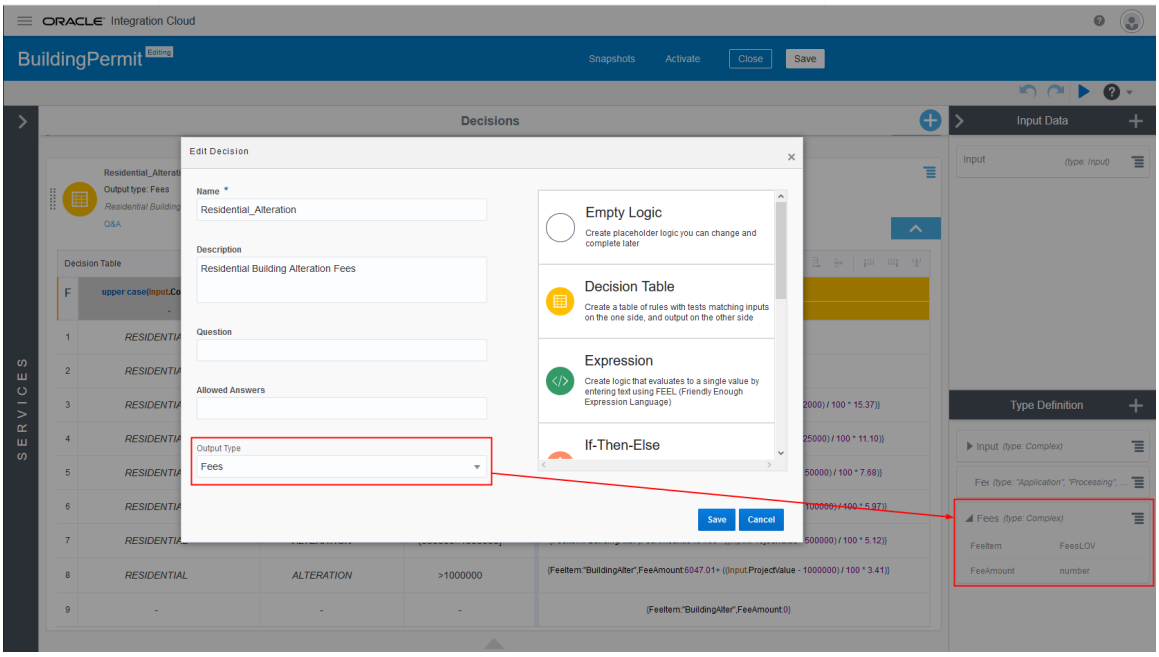


3. (Required) Add the output data type name to every decision and use allowed values in the **Decision Table** grid.

   To edit a decision, click the decision menu button and select **Edit**.

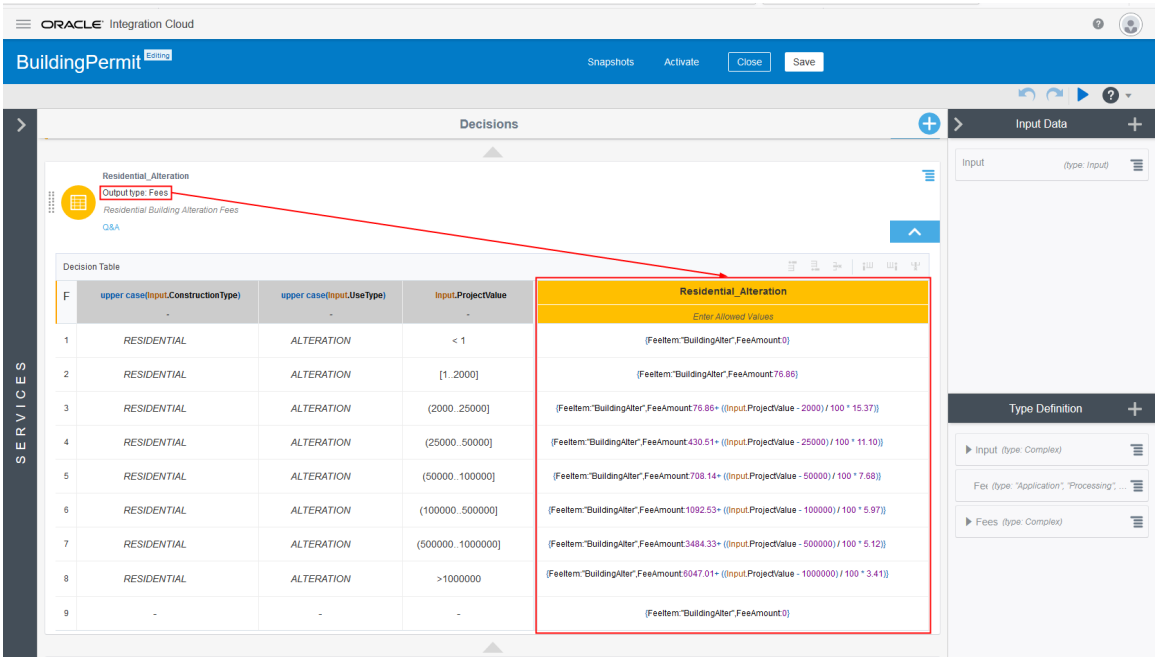
   Select the defined output type from the **Output Type** list.

   This example illustrates the output data type name *Fees* added to the Residential_Alteration decision. The output type provides constraints on which values are allowed in the decision model.



This is an example of the final decision values based on the output type.

The decision output type in the Residential_Alteration decision is *Fees*. The validation ran on the Fees values entered in the Decision Table, and no errors were returned.



4. (Required) You must also configure the services in the **Services** panel to the left of the decisions, and activate the model. For more details, see the documentation for OIC, *Creating a Service*.

When the Oracle Public Sector Compliance and Regulation service submits a request to the OIC, after running the request against the decision model, the application returns the fee item name and the fee item amount.

Related Topics

- Setting Up Fee Items
- Setting Up Fee Schedules

3 Setting Up GIS

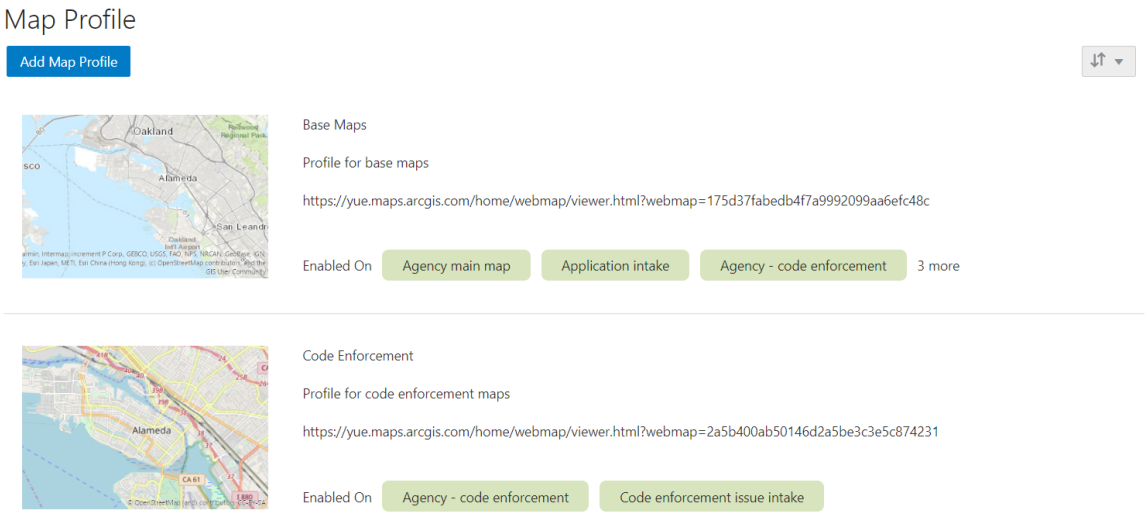
Setting Up Map Profiles

Use map profiles to configure specific instances of map functionality in the system. Profiles set the default extent of the map (the area shown by default) as well as controlling the availability of certain map options.

A profile can be linked to multiple maps. All of a profile's maps share the same map service URL, default map extent, and default base map, but the maps have individual configuration options to control end-user options for the map.

Every map must be associated with at least one map profile. If a map has multiple profiles, a **Switch Map Profile** button appears on the map toolbar, and end users can choose which profile to use.

This example illustrates the Map Profile list page. Each row includes a thumbnail map, lists up to three maps that are associated with the profile, and states how many additional maps (if any) are associated with the profile.



Prerequisite

Mapping capabilities depend on integration with a map service such as Esri's Geographic Information Systems (GIS). Before you set up map profiles, publish your map service so that it can be referenced from within the Oracle system.

Delivered Maps

The following tables describe the delivered maps in the Oracle Public Sector system.

> **Note:** Make sure that every map you use is associated with a profile.

This table lists the main maps that display the agency's permits, planning applications, and projects. There are separate maps for different users. For more information, see *Using the Main Map and Explore Your City Map*.

Map Name	Description
Agency main map	The main map that appears when an agency staff member clicks the **Main Map** icon in the page header.
Guest/anonymous user map	The main map that appears when an anonymous user clicks the **Explore Your City** tile on the landing page.
Registered public user map	The main map that appears when a public user who is signed in clicks the **Explore Your City** tile on the landing page. Registered users and anonymous users see different maps because registered users have additional options for starting permit and planning applications and for limiting searches to just the user's own applications. These options are not configured on the map profile, but the existence of separate maps for public users and anonymous users means that you must configure them separately.

This table lists additional maps that support permits and planing applications:

Map Name	Description
Application intake	The property picker map. This map appears in a modal window that is accessed from an application intake form or, for submitted applications, from the application details Property Information page. See *Working with Property Information*.
Mobile inspection	The map that inspectors see when using mobile inspection functionality for permits and planning applications. See *Oracle Inspector Overview for Permits*.
Permit list	The map that appears on the application list pages when they are in map view. The list page for agency staff is the Transactions page. This page includes lists for both permits and planning applications. The same map is used for both. The list page for registered public users is the Applications page. These list pages all include List View and Map View buttons for toggling between views. See *Managing Transactions*.
Public notification	The map that appears on the Public Notification page. Agency staff uses this page to define a notification area. See *Creating Map-Based Public Notification Lists*.

This table lists additional maps that support code enforcement:

Map Name	Description
Agency - code enforcement	The map that code enforcement technicians can view from certain tabs on the Worklist page. This map shows the location of issues that appear on the selected worklist tab. See *Using the Worklist*.
Code enforcement issue intake	The map that appears on the page where public users who are reporting a code enforcement issue identify the location of the issue. See *Reporting Issues*.
Mobile code enforcement inspections	The map that inspectors see when using mobile inspection functionality for code enforcement. See *Oracle Municipal Code Officer Overview*.
Public - code enforcement	The map where public users can view recent code enforcement issues. See *Viewing Recent Issues*.

This table describes the map that appears in transaction headers:

Map Name	Description
Transaction header	The map that appears at the top of various detail pages. For example, this map appears in the detail pages for permits, planning applications, incidents, cases, parcels, addresses, and owners. **Note:** Link this map to exactly one map profile. Profile switching is not allowed for this map.

Note: If your agency defines a new map page, the new map must be added to the Map Page lookup (ORA_PSC_COM_SYS_MAP_PROFILE) so that it is available to be associated with a map profile.

Adding a Map Profile

1. Select **GIS Setup** > **Map Profile.**

   If no profiles exist, the Map Profile Details page opens so that you can create the first profile. If at least one profile exists, the Map Profile list page appears.
2. If the Map Profile list page appears, click **Add Map Profile.**

   The Map Profile Details page appears.
3. Enter the following basic profile information:

Page Element	Description
Profile ID	Enter a unique identifier for the map profile.

Page Element	Description
Profile Name	Enter a descriptive name for the map profile. This identifies the profile on the Map Profile list page. When end users view a map with multiple profiles, they can switch profiles by choosing from a list that displays this name along with a map thumbnail.
Description	Enter a more detailed description of the map profile. This also appears on the Map Profile list page to help identify the profile.

4. Enter the URL for your map service in the **Map Service URL** field.

   The URL to the profile provides default values for the map extent, which you will confirm or change later in this procedure.
5. If the **Base Map** field is editable, choose the type of map to display.

   The Esri server settings control whether this field is editable.

   The options are *Dark gray canvas, Light gray canvas, Imagery with labels, National Geographic, Topographic, Open Street Map, Imagery, Streets, Terrain with labels,* and *Oceans.*

   When you link specific maps to the profile, you will configure whether users can change the map type.
6. Set the map extent.

   The map extent defines the geographical area that the map initially displays. When you create a new map profile, a generic map illustration appears above the **Choose Map Extent** option. After you choose the map extent, a preview thumbnail of your actual map extent replaces the generic illustration.

   Although the map service URL provides a default map extent, you still need to click **Choose Map Extent** to load the default extent into the profile and optionally modify it.

   > **Note:** Maps in transaction headers are initially centered on the transaction location. Therefore, the transaction header map uses the default extent from the map profile only if the transaction is not associated with a specific location.

   a. Click **Choose Map Extent.**

      The Choose Map Extent page appears. The map service URL that you previously provided sets the default map extent, and the page displays a map with that default extent.
   b. If necessary, modify the default map extent supplied by the map service URL.

      Oracle provides the ability to easily set a new map extent without making any changes to the GIS service. To change the extent, pan and zoom until you can see the desired extent, then use the **Choose Map Extent** toolbar button to draw a selection rectangle. This sets the new extent.

      The following fields describe the map extent by identifying a coordinate system and listing the minimum and maximum X and Y values on the coordinate system.

Page Element	Description
X-Min of Default Map View	The top-left X-coordinate of the initial map extent.
X-Max of Default Map View	The bottom-right X-coordinate of the initial map extent.
Y-Min of Default Map View	The bottom-left Y-coordinate of the initial map extent.
Y-Max of Default Map View	The top-right Y-coordinate of the initial map extent.
Spatial Reference	The geographic coordinate system or map projection used by the mapping service to display the map. The map service URL that you previously supplied sets this value.

c. Click **OK** to close the Choose Map Extent window.
The thumbnail map on the Map Profile Details page is updated to match your map extent.

7. Link maps to the profile and define settings for each map:
   a. Click **Add Map Page.**
   b. In the **Map Page** field, select a map to link to the profile.
   The maps are *Agency main map, Application intake, Agency - code enforcement, Code enforcement issue intake, Mobile code enforcement inspections, Public notification, Transaction header, Guest/anonymous map, Mobile inspection, Permit list, Registered public user map,* and *Public - code enforcement.*
   These maps are described in detail above.
   c. Configure map-specific options.
   Depending on the map that you are configuring, some map options might not be available to enable or disable. For example, you cannot enable selection tools or window docking on the maps for mobile devices, and zoom tools are the only widgets available in the transaction header map.
   If you create any custom maps, there are no restrictions on which widgets you can enable, so take extra care when configuring those maps.
   Use these fields to configure map options:

Page Element	Description
Enable Zoom	Indicate whether the map toolbar includes **Zoom In** and **Zoom Out** tools.
Enable Default Map View	Indicate whether the map toolbar includes the **Show Default Map View** tool. This tool restores the map to its initial extent after a user zooms or pans to change the display area.

Page Element	Description
Enable Base Map Gallery	Indicate whether the map toolbar includes the **Select Base Map** tool. This tool lets users change the base map from the one specified in the profile. For example, if the profile's base map is topographic, users can change to a map with satellite imagery.
Enable Map Layers	Indicate whether the map toolbar includes the **Select Layers** tool and, depending on your GIS configuration, the **Identify GIS Information** tool. The **Select Layers** tool lets users see the list of layers and switch layer visibility on and off. Examples of layers include environmental, zoning, or infrastructure information provided by the map service. The **Identify GIS Information** icon gives users the ability to click map objects such as parcels to display a pop-up window with object information. This option is available if the GIS administrator has configured the GIS service to provide this information.
Enable Selection Tools	Indicate whether the map toolbar includes the **Show/Hide Selection Tools** icon. Clicking this icon opens a separate toolbar with tools for selecting and deselecting parcels on a map. The ability to select parcels on the agency main map, the registered public user map, and the guest/anonymous user map enables users to view associated transactions. Registered users and agency staff can additionally start an application for selected parcels. The ability to select parcels on the public notification map enables users to create a notification area around the selected parcels.
Enable Detail Window Docking	Indicate whether the map detail window is docked to the side of the view. The detail window is the pop-up window that appears when a user clicks a map marker or other GIS feature such as a parcel.
Detail Window Dock Position	Specify the position where the map detail window is initially docked: *Auto, Bottom left, Bottom center, Bottom right, Top left, Top center,* or *Top right.* This field is relevant only if you enable detail window docking. This field does not apply to mobile devices, where the detail window always appears at the bottom of the screen.

d. Click **Save** to close the Add Map Page window.

8. If necessary, click a linked map to re-open the Add Map page.

- To edit settings, make your changes and then click **Done.**
- To remove the map from the profile, click the **Delete** button.

9. Click **Save** to save the map profile.

Modifying a Map Profile

1. Select **GIS Setup** > **Map Profile** .
2. On the Map Profile list page, click the row for the profile that you want to modify.
3. Update the settings as needed.
4. Click **Save.**

Deleting Map Profiles

To delete a map profile:

1. Select **GIS Setup** > **Map Profile.**
2. On the Map Profile list page, click the row for the profile that you want to delete.
3. On the Map Profile Details page, click **Delete.**

Setting Up GIS Attribute Mapping

Use Global Information Systems (GIS) attribute mapping to specify information about your map service parcel layer.

Prerequisites

Before you enter the information about your map service layers, you must:

- Publish the map service, which must have parcel, address, and owner layers.

  When you save the URL for a map service layer, an error message appears if the layer is not available.
- Ensure that the parcel layer has a field with parcel IDs that match the parcel IDs in the Oracle system.

  Parcel IDs must match exactly, with no formatting differences.

Setting Up the Service Layer URLs

To set up the layer service URLs:

1. Select **GIS Setup** > **Attribute Mapping.**

   The GIS Attribute Mapping page appears.
2. Enter parcel layer information:

Page Element	Description
Parcel Layer Service URL	Enter the URL for your parcel layer feature service.

Page Element	Description
	The URLs for the different layers of an Esri map service have numeric identifiers. The URL that you enter here ends with the number for the parcel layer as in the example *https://servername/arcgis/rest/services/Your_City/MapServer/4* You must publish your parcel layer feature service before you enter the URL here.
Parcel Number	Select the parcel layer GIS attribute that provides the unique identifier for each parcel. The values in the drop-down list come from the parcel layer that you specify. Select the GIS attribute that provides the same identifiers that are used in the parcel table in the Oracle system. For information about setting up the parcel table, see *Setting Up Parcels*. On maps used as property pickers, clicking a parcel on a map retrieves the parcel identifier from the map service. This value is used as criteria for searching the Parcel table, and the search results appear in a modal window. As long as the same parcel number exists in the Parcel table, the search results include just one value, representing the selected parcel.

3. Enter address layer information:

Page Element	Description
Address Layer Service URL	Enter the URL for your address layer feature service. The URL ends with the number for the address layer. You must publish your address layer feature service before you enter the URL here.
Parcel Number	Select the address layer GIS attribute that provides the unique identifier for each parcel.

4. Enter owner layer information:

Page Element	Description
Owner Layer Service URL	Enter the URL for your owner layer feature service. The URL ends with the number for the owner layer. You must publish your owner layer feature service before you enter the URL here.
Parcel Number	Select the owner layer GIS attribute that provides the unique identifier for each parcel.

5. Enter boundary layer information:

Page Element	Description
Boundary Service Layer URL	Enter the URL for your boundary layer feature service. The URL ends with the number for the boundary layer. This layer identifies the agency's boundaries so that the system can check whether a location on a map is within those boundaries. For example, in the code enforcement system, issue locations must be within the agency's boundaries. Your GIS administrator must create and publish your boundary layer feature service before you enter the URL here.

6. Click **Save.**

Setting Up Access to Secure Map Services

This topic discusses how to set up access to the agency's secure private maps.

Overview of Secure Map Access

Giving users access to non-public maps (maps that can't be accessed directly through a browser) involves setup in both the Esri and Oracle systems:

1. In the Esri system, set up proxy users with access to the maps.

   You can set up proxy users with access to one or multiple secure map services.
2. In Oracle, create a secure access definition that includes the user ID and password for the proxy.

   The secure access definition also includes a URL for the map service to be accessed and a URL for the web service that generates an authentication token for accessing the map services.

On the Oracle side, create separate secure access definitions for each of the map service URLs that you need to access. You can create definitions for specific map services, such as *https://portal.city.net/arcgis/rest/services/TaxParcels/MapServer,* or for generic services such as *https://portal.city.net/arcgis/rest/services.*

When an Oracle user attempts to access a secured map, the authentication process checks first for access information for the specific map service, then for access information for the generic service.

Prerequisites

To enable access to secure map services, you must:

1. Set up proxy users in the Esri system.
2. Give these proxy users appropriate access to the secure maps services that will be accessed from the Oracle system.

Setting Up Secure Map Access

To set up secure map access:

1. Select **GIS Setup** > **Secure Map Access.**

   The GIS Secure Map Access page appears.
2. Click the **Add** button to create a new secure access ID.
3. Enter the following information:

Page Element	Description
Secure Access ID	Enter an identifier for this configuration.
Description	Enter a description for this configuration
GIS Service URL	Enter the URL to the secured map service that will be accessed with this configuration.
Token Service URL	Enter the URL to the web service that generates an authentication token for accessing secure map services.
User Name	Enter the user name for the proxy user that is used to access secure map services.
Password	Enter the password that the proxy user uses to access secure map services.
Confirm Password	Re-enter the password. The re-entered password is compared to the originally entered password to help catch data entry errors.

4. Click **Save.**

   Saving tests the access information that you provided. If the test fails, you aren't able to save.

Modifying a Map Profile

1. Select **GIS Setup** > **Secure Map Access.**

   The GIS Secure Map Access page appears.
2. On the GIS Secure Map Access list page, click the row for the profile that you want to modify.
3. Update the settings as needed.
4. Click **Save.**

Deleting Map Profiles

To delete a map profile:

1. Select **GIS Setup** > **Secure Map Access.**

The GIS Secure Map Access page appears.

2. On the GIS Secure Map Access list page, click the row for the profile that you want to delete.
3. On the GIS Secure Map Access detail page, click **Delete.**

4 Configuring Oracle Intelligent Advisor

Overview of Oracle Intelligent Advisor Configuration

This topic provides an overview of how Oracle Intelligent Advisor is used within Public Sector Compliance and Regulation and how it is configured.

If your site already has an installation of Oracle Intelligent Advisor, you can integrate its functionality with the permits service. The policy models created in Oracle Intelligent Advisor can act as the logic models running behind questionnaires that public users fill out to determine which permits they need to apply for depending on the nature of the project they are planning.

The topics in this chapter describe the setup pages that an administrator would view and use to configure the mapping of metadata between the permits service and the Oracle Intelligent Advisor application.

> **Note:** Most of the data displayed on the pages within the Policy Modeling menu folder are read-only. In the current release, you do not create or update the definitions such as entity models, metadata models, or enumerations. These definitions are provided for your implementation in the form of delivered seed data. This documentation is provided to describe the visible information related to the policy modeling feature. Only the Metadata Models page contains fields that can be updated.

Setting Up Entity Models

This topic describes the settings used to configure entity models used when implementing Oracle Intelligent Advisor for use with the permits application.

Adding an Entity Model

> **Note:** Most of the data displayed on the pages within the Policy Modeling menu folder are read-only. In the current release, you do not create or update the definitions such as entity models, metadata models, or enumerations. These definitions are provided for your implementation in the form of delivered seed data. This documentation is provided to describe the visible information related to the policy modeling feature. Only the Metadata Models page contains fields that can be updated.

1. Select **Policy Modeling** > **Entity Models** .
2. Click **Add.**
3. On the Entity Model Details page, enter these values:

Page Elements	Description
Name	Enter a name to identify the model within the application.

Page Elements	Description
Description	Provide additional information regarding the purpose of the model.
Enabled	Use to enable or disable a model by turning the control on or off.

4. Click **Add** in the Entities grid.
5. On the Entity Details page, enter these values:

Page Elements	Description
Name	Name of the entity.
Description	Additional information to identify the entity and describe its purpose.
Hidden from Policy Modeling	If set to true, then this entry will not be present in the Get MetaData response to Oracle Intelligent Advisor.
Top-Level Entity	Indicates if the object is the highest level entity object.
Policy Modeling Name	The functional name for an entity or attribute as it appears within Oracle Intelligent Advisor.
Use as Mapped in Entity	Defines if the entity object can be selected as an input entity.
Use as Mapped Out Entity	Determines if the entity object can be selected as an output entity.
Parent Entity Name	The name of the parent entity object of a child object.
Cardinality with Parent Entity	Indicates the cardinality relationship with the parent entity object, such as one-to-one, one-to-many, many-to-one, or many-to-many.
Policy Modeling Relationship Name	The name of the relationship between two entities as it appears in Oracle Intelligent Advisor.
Supports Attachment	Determines if attachments can be collected for rows of the entity object.

6. Click **Add** in the Entity Attributes grid to add attributes for the entity.
7. On the Entity Attribute Details page, enter these values:

Page Elements	Description
Name	The system name of the entity attribute.
Data Type	The data type of the attribute as it is defined in Oracle Intelligent Advisor. For example: o java.lang.String o java.lang.Long
Primary Key	The primary key of the underlying view object.
Policy Modeling Name	The functional display name for an entity or attribute as it appears in Oracle Intelligent Advisor.
Hidden from Policy Modeling	If set to true, then this entry will not be present in the Get MetaData response to Oracle Intelligent Advisor.
Mandatory	Determines if the field *must* be mapped from an attribute in a policy model.
Policy Modeling Data Type	Describes the data type of the field defined in Oracle Intelligent Advisor. It must be specified if no enumeration-type attribute is provided, and it cannot be specified if an enumeration-type attribute is provided. Options are: o String o Boolean o Decimal o Date o Date-time o Time-of-day
Use as Mapped In Attribute	Determines if the field can mapped from an attribute for the purpose of submitting data.
Use as Mapped Out Attribute	Determines if the field can mapped from an attribute for the purpose of submitting data.
Default Value	Enter a default value for this attribute. If added, the application includes the value in the load response to Oracle Intelligent Advisor.

Page Elements	Description
Enumeration Name	Specifies the ID of the enumeration that defines a field's data type.

8. Click Save.
9. Click Save on the Entity Details page.
10. Click Save on the Entity Model Details page.

Setting Up Metadata Models

This topic describes how to set up Oracle Intelligent Advisormetadata models and define entity relationships.

> **Note:** Most of the data displayed on the pages within the Policy Modeling menu folder are read-only. In the current release, you do not create or update the definitions such as entity models, metadata models, or enumerations. These definitions are provided for your implementation in the form of delivered seed data. This documentation is provided to describe the visible information related to the policy modeling feature. Only the Metadata Models page contains fields that can be updated.

To set up Oracle Intelligent Advisor metadata models:

1. Select **Policy Modeling** > **Metadata Models** .
2. Click Add for the Metadata Models grid.
3. On the Metadata Models Details page, add enter these values:

   > **Note:** You can update the following fields for a metadata model definition: **Supports Policy Modeling Checkpoints, Anonymous Users Can Save Data,** and **Active Model.** By default these fields are turned-off. You can turn them on according to your business requirements.

Page Element	Description
Name	Enter the functional display name of the metadata model.
Description	Provide additional description to identify the purpose of the metadata model.
Supports Policy Modeling Checkpoints	Turn on to indicate that the metadata model is designed to support checkpoints.
Anonymous Users Can Save Data	Turn on to enable the anonymous (non-signed-in user) to save data.
Active Model	Turn on to activate or deactivate the model.

4. Click Add for the Metadata Entity Relationships grid, and enter these values:

Page Element	Description
Name	Enter the entity relationship name.
Mark as Global Entity	Turn on if the entity is global.
Cardinality with Global Entity	Indicate the cardinality with the global entity (one-to-many, many-to-one, and so on).
Policy Modeling Relationship Name	The name of the relationship between two entities as it defined within in Oracle Intelligent Advisor.

5. Click Add for the Metadata Entity Links grid, and enter these values:

Page Element	Description
Source Entity Policy Modeling Name	Represents the policy modeling name for the entity in the source entity model.
Target Entity Model Name	Enter the target entity model.
Target Entity Policy Modeling Name	Represents the policy modeling name for the entity in the target entity model for this link.
Description	Provide any additional details to describe the purpose of metadata entity link.
Cardinality with Target Entity	Indicate the cardinality with the target entity (one-to-many, many-to-one, and so on).
Policy Modeling Relationship Name	The name of the relationship between two entities as it appears in Oracle Intelligent Advisor.

6. Click **Save.**
7. Click **Save** on the Metadata Entity Relationship Details page.
8. Click **Save** on the Metadata Model Details page.

Setting Up Enumerations

This topic describes how to configure enumerations for policy modeling. An *enumeration* is a tool for managing lists of potential values for a non-boolean attribute in your policy model. Enumeration are also referred to as value lists.

Note: Most of the data displayed on the pages within the Policy Modeling menu folder are read-only. In the current release, you do not create or update the definitions such as entity models, metadata models, or enumerations. These definitions are provided for your implementation in the form of delivered seed data. This documentation is provided to describe the visible information related to the policy modeling feature. Only the Metadata Models page contains fields that can be updated.

To add an enumeration:

1. Select **Policy Modeling** > **Enumerations** .
2. On the Enumerations page, click **Add.**
3. On the Enumerations Details page, enter these values:

Page Element	Description
Enumeration Name	The functional display name of the enumeration.
Enumeration Type	The data type of the enumeration, such as: o String o Number o Boolean o Time
Description	Provide additional information to help describe the purpose of the enumeration.
Child Enumeration Name	Specify the name of a linked child enumeration, as needed.

4. Click **Add** for the Enumeration Values grid.
5. On the Enumeration Value Details page, enter these values:

Page Element	Description
Enumeration Value	Enter the function display name of the enumeration value.
Description	Provide additional information to describe the purpose of the enumeration value.

6. Click **Add** for the Child Enumeration Values grid for any child enumeration values.

Page Element	Description
Child Enumeration Value	Enter the function display name of the child enumeration value.
Description	Provide additional information to describe the purpose of the child enumeration value.

Page Element	Description

7. Click **Save.**
8. Click **Save** on the Enumeration Value Details page.
9. Click **Save** on the Enumeration Details page.

Mapping Enumerations to Metadata Models

This topic describes how to map defined enumerations to existing metadata models.

> **Note:** Most of the data displayed on the pages within the Policy Modeling menu folder are read-only. In the current release, you do not create or update the definitions such as entity models, metadata models, or enumerations. These definitions are provided for your implementation in the form of delivered seed data. This documentation is provided to describe the visible information related to the policy modeling feature. Only the Metadata Models page contains fields that can be updated.

To map an enumeration to a metadata model:

1. Select **Policy Modeling** > **Enumeration Mapping.**
2. Click **Add** for the Metadata Models grid.
3. On the Metadata Models Details page, enter the name of the existing metadata model to which you want to map the enumeration.

   > **Note:** Once added, the remaining attributes, such as Supports Policy Modeling Checkpoints, are read from the metadata model definition.

4. Click **Add** for the Metadata Entity Relationships grid.
5. On the Metadata Entity Relationship Details page, enter these values:

Page Element	Description
Name	Enter a name for the relationship.
Mark as Global Entity	Indicates if this is a global entity for this metadata model.
Cardinality with Global Entity	Specify the cardinality relationship that this entity has with the global entity identified for this metadata model, such as one-to-one, one-to-many, and so on.
Policy Modeling Relationship Name	The name of the relationship between two entities as it appears in Oracle Intelligent Advisor.

6. Click **Add** for the Metadata Entity Links grid.
7. On the Metadata Entity Link Details page, enter these values:

Page Element	Description
Source Entity Policy Modeling Name	Enter the policy modeling name for the entity in the source entity model for this link.
Target Entity Model Name	Enter the target entity model for this link.
Target Entity Policy Modeling Name	Enter the policy modeling name for the entity in the target entity model for this link.
Description	Provide any additional information to describe the purpose of this metadata entity link.
Cardinality with Target Entity	Specify the cardinality relationship that this entity has with the target entity model name identified for this metadata link, such as one-to-one, one-to-many, and so on.
Policy Modeling Relationship Name	The name of the relationship between two entities as it appears in Oracle Intelligent Advisor.

8. Click **Save.**
9. Click **Save** on the Metadata Entity Relationship Details page.
10. Click **Save** on the Metadata Model Details page.

Related Topics

- Setting Up Metadata Models
- Setting Up Enumerations

Managing Proxy Users

This topic describes how to manage proxy users for enabling integration between Oracle Intelligent Advisor and your Public Sector Compliance and Regulation service.

Oracle Intelligent Advisor connects to your Public Sector Compliance and Regulation service through a provided web service connector named pscOpaWSConnector.

This connector requires proper WS-Security credentials to handle the transactions between the Public Sector Compliance and Regulation service and the Oracle Intelligent Advisor service.

When configuring the connection within the Oracle Intelligent Advisor hub, in the WS-Security section of the New Connection page, a user ID and password is required.

The user ID entered must have the following role within their role hierarchy:

PSC Oracle Intelligent Advisor Proxy User (ORA_PSC_OPA_PROXY_USER_DUTY)

This duty role contains the following privilege:

Access Oracle Intelligent Advisor Web Service Connector Privilege (PSC_OPA_WSC_PRIV)

This privilege allows the proxy user to integrate Oracle Intelligent Advisor with your Public Sector Compliance and Regulation service.

By default, the delivered SYSTEM_ADMIN has the PSC System Administrator job role, which inherits the PSC Oracle Intelligent Advisor Proxy User duty role. Any custom (cloned) role or created user must have PSC Oracle Intelligent Advisor Proxy User duty role if you intend to use that user ID as the proxy user for the Oracle Intelligent Advisor WS-Security credentials.

Related Topics

- Working with Roles in the Security Console
- Managing Roles in Public Sector Compliance and Regulation

Managing the Oracle Intelligent Advisor Hub Endpoint

Administrators set up the Public Sector Compliance and Regulation services that are required to integrate with the Oracle Intelligent Advisor using the topology entries.

To set up the Oracle Intelligent Advisor Hub endpoint:

1. Select the **Setup and Maintenance** tile on the Agency Springboard. On the Setup page, select the offering: Public Sector Permits or Public Sector Planning and Zoning and then select the OPA Questionnaire functional area.
2. On the right panel, select the task named Manage OPA Hub Endpoint to open the setup page.
3. Expand the Server Details section and fill up the following fields, with the values you received while setting up Oracle Intelligent Advisor.

Page Elements	Description
Server Protocol	Select the protocol of the Oracle Intelligent Advisor service.
External Server Host	Enter the host of the Oracle Intelligent Advisor service.
External Server Port	Enter the port of the Oracle Intelligent Advisor service.

4. Click **Save and Close**.

Managing the Oracle Intelligent Advisor Hub

Administrators set up the Oracle Intelligent Advisor hub that is required to integrate with Public Sector Compliance and Regulation services.

Oracle Intelligent Advisor Setup for Integrating with Public Sector Compliance and Regulation

This is a two-step process:

1. Authorizing Embedded Interviews
2. Creating Connections

Authorizing Embedded Interviews

1. Log in to the Policy Automation Hub web interface with the user credentials of *Deploy Admin*.
2. Click the Permissions tile to open the Permissions page. Click the menu in the right top of the page and select **Access Settings**.
3. On the Access Settings page, click **Add Host** under Interview Access Control.
4. In the CORS Hosts field, enter the Public Sector Compliance and Regulation application host address.
5. Click **Apply**.

Creating Connections

1. Log in to the Policy Automation Hub web interface with the user credentials of *Deploy Admin*.
2. Click the **Connections** button on the banner to open the Connections page.
3. On the Connections page, click the Actions drop-down menu and select Create a new Connection option to open the New Connection page and enter values for the various fields:

Page Element	Description
Name	Enter a name for the connection.
Type	Select **Web service**.
Collection Access	Select the collection that you have created, to gain access to the connection. Click **Allow**. The default value for this field is **Default Collection**. You can include any additional collections that you want to allow access to.
URL	Enter the URL of the connector, which is deployed with other services – the FSCM base URI from the topology manager. Append the below string to the URL of the connector as shown here: <FSCM base URI>/fscmPojoService/pscOpaWSConnector?MDMN=OPAResult
Use Custom Certificate (optional)	Select to use a custom certificate defined in Policy Automation Hub. These custom certificates will be recognized by outbound *https* calls made by a Policy Automation site. If not selected, the connection will only trust the built-in root certificates.
Version	Select the following web service version:

Page Element	Description
	12.2.13
SOAP ActionPattern (optional)	Specify the *soap:operation soapAction* name expected by the web service.

OAUTH for Data Operations:

Page Element	Description
Provide OAUTH bearer token in HTTP header on Load and Save actions	Select to allow you to enter a URL parameter and enter the value *jwt* in the URL Parameter field. The token's value is passed by specifying the parameter in the query string of the interview's start session URL. This value is then passed to the Web Service connector as an **OAuth 2.0 HTTP Authorization** header whenever a Load or Save request is sent.

WS-Security:

Page Element	Description
Provide WS-Security Username token in SOAP actions.	Select the option to allow you to enter values for the fields in the section.
Applies to	Select applies to **All**.
Username	Enter a username for the purpose of connecting securely to the web service. Note that this is not related to the username of the logged-in Policy Automation Hub user. If you have installed Oracle Intelligent Advisor, then as part of the Fusion Onboarding process you must have created a user having the following Oracle Intelligent Advisor proxy user Duty role: **ORA_PSC_OPA_PROXY_USER_DUTY**. Use the same user name in this field.
New Password	Enter a password.
Include timestamp with a 5 minute expiration (optional).	Select to include a timestamp with a validity of 5 minutes. Note: The web service connector time must be synchronized to the Oracle Intelligent Advisor server.

4. Click **Save and Close** to complete the process of creating a new connection.

Managing Oracle Intelligent Advisor Policies for your Agency

This topic describes how to set up Oracle Intelligent Advisor policies for your Agency.

You can enter an Oracle Intelligent Advisor policy model at the agency level or at the offering level (for example, for the Permits offering). When the policy is used by a specific offering, the offering-specific policy model takes priority over the agency-level policy model.

> **Note:** To identify a policy model, you enter the deployment name listed in the Deployment page. The Deployment page is where the deployment and activation of policy models is managed. To access the Deployment page, log in to the Policy Automation Hub web interface with a user role of *Policy Author* or *Deploy Admin*. On the Dashboard page, click the deployments tile to open the Deployments page. From the list of all projects currently deployed, select the desired deployment name.

To define the Oracle Intelligent Advisor policies for your agency:

1. Select **Common Setup** > **Agency** .
2. Click a row on the Agency Information tab.
3. To define an agency-level policy model:
    a. Enter the policy information in the **Oracle Policy Automation ID** field on the Agency Information tab.
    b. Click **Save.**
4. To enter an offering-level policy model:
    a. Click the Features tab.
    b. Click the **Options** link for the offering you are configuring.
    c. On the Permit Options page, enter the policy information in the **Oracle Policy Automation ID** field.
    d. Click **Save.**

> **Note:** You must repeat the steps outlined in this topic and in the *Setting Up Metadata Models* topic when you are moving the content from the Test environment to your Production environment.

5 Setting Up Additional Integrations

Setting Up Contractor Integration

This topic describes how to import and activate integration settings that enable you to validate contractor information against data from the contractor licensing body.

Oracle provides pre-built integrations with specific contractor licensing bodies. To set up this integration, download the integration files from My Oracle Support and then import them into Oracle Autonomous Integration Cloud (OIC). To use an integration after you set it up, use the Contractor License Options page. For more information, see . *Setting Up Contractor License Options*.

To set up contractor integration:

1. Go to My Oracle Support (MOS) and access Doc ID 2672514.1.
2. Follow the instructions on the MOS page, and save the .iar integration file for your contractor licensing body.
3. Access the main console in OIC.
4. Click **Integrations** in the left frame.
5. Click the **Import** button at the top of the Integrations page.
6. Select the integration file for your licensing body, then click the **Import** button.
7. Click **Connections** in the left frame.
8. Test the **Generic REST Trigger** connection:
    a. Access the detail page for the **Generic REST Trigger** connection..
    b. Click the **Test** button.
    c. When the test is 100% complete, click **Save** and then **Close.**
9. Test the integration for the imported licensing body:
    a. Access the detail page for the licensing body connection.
    b. Click the **Test** button.
    c. When the test is 100% complete, click **Save** and then **Close.**
10. Click **Integrations** in the left frame to return to the Integrations page.
11. Turn on the activation switch for the new licensing body integration.
12. In the Activate Integration window that appears, click the **Activate** button.

Setting Up a Proxy Role and User for Integrated Voice Response

Set up a proxy user in the Oracle Security Console to give your Integrated Voice Response (IVR) system access to PSCR.

The IVR system accesses permit information via REST, using the proxy user credentials. During a call, a public user provides the IVR system with both a permit number and their personal IVR code. The IVR system sends that code along with any request to access the permit information in the PSCR system. PSCR verifies that the code matches the IVR code that's stored in the permit owner's account. If the code matches, the request is honored.

In this procedure you will use the Security Console to:

1. Create a custom role for the IVR access.
2. Assign a delivered duty role to the custom role.
3. Create the IVR proxy user.
4. Assign the custom IVR role to the IVR proxy user.

For more information about using the Security Console, see: *Using the Security Console*.

Creating the IVR Custom Role

To create the PSC IVR Proxy User role:

1. Navigate to the Security Console.

   To navigate to the Security Console, you have these options:

   - In Functional Setup Manager, click the task: *Create Integrated Voice Response Proxy User.*
   - Click Setup and Maintenance on the Agency Springboard, and on the Fusion Applications home page, select **Navigator** > **Tools** > **Security Console.**
2. Select the Roles tab.
3. Click **Create Role.**
4. On the Create Role: Basic Information page enter the following:

Page Element	Value
Role Name	*PSC IVR Proxy User*
Role Code	*CUSTOM_PSCR_IVR_PROXY_USER*
Role Category	*Financials — Job Roles*

5. Click the Role Hierarchy step, and add the following duty role: *PSC Interactive Voice Recognition Proxy User (ORA_PSC_IVR_PROXY_USER_DUTY).*
6. Click the Summary step and click **Save and Close.**

Creating the IVR User

To create the IVR proxy user:

1. In the Security Console, click the Users tab.
2. On the User Accounts page, click **Add User Account.**
3. On the Add User Account page in the User Information section, enter a **Last Name** and **User Name** of your choice.

   > **Note:** The name given for the proxy user should be generic, such as *IVR Proxy User.*

4. Enter a **Password** of your choice and confirm it.
5. Click **Add Role** for the Roles grid, and assign this role to your proxy user:

- Role Name: *PSC IVR Proxy User*
- Role Code: *CUSTOM_PSCR_IVR_PROXY_USER*

6. Click **Save and Close.**