



Oracle Integration Cloud Process Automation Apps

Integrate, Automate and Analyze

ORACLE LAB BOOK | MARCH 2018




By
Nathan Angstadt

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Disclaimer	1
Workshop Introduction	3
Scenario	4
Lab Overview	5
Pre-Requisites	6
 Section 1: A Warranty Claim Application	7
Section Objectives	7
Examine a Process App	7
Run a Process App from Workspace	8
 Section 2: Warranty Claim Application User Experience	10
Section Objectives	10
Embed a Process App in a Website	10
Run an Embedded Process App from a Web Site	14
 Section 3: Building a Product Registration Application	16
Section Objectives	16

Create a Process App from QuickStart	17
Import Integration Connections	19
Build a Web Form	21
Build a Process Flow	31
Deploy a Process App for Testing	35



Section 4: A Complete Warranty Claim Business Process 37

Section Objectives	37
Add your Product Registration Application to the Customer Portal	37
Register a Product	38
Make a Warranty Claim	39
Close a Service Cloud Incident	40
Gain Insight into the Warranty Claim	41

Workshop Introduction

This workshop will provide you hands on experience building Process Apps; business applications that digitize, automate and streamline end to end business processes. But why are these applications important?

SaaS and on-premise CX, HCM and ERP enterprise applications form the pillars of business operations. However, many business processes include activities and stakeholders that fall outside of the enterprise application scope or user community. And even more business processes span multiple enterprise applications, creating disconnects as the cross-pillar business process transitions from one enterprise application to another. In many businesses, it is common that these business processes have evolved into an inefficient set of ad-hoc activities managed by emails, spreadsheets and lots of manual repetitive tasks. The result is ineffective operations, lack of governance and disconnected customer and employee experiences. These threats to a competitive business are opportunities for digital transformation.

The Oracle Integration Cloud platform makes it easy to build, deploy and run business applications that digitize, automate and streamline these business processes, integrate the pillar enterprise applications and provide real-time insight into business operations. The ability to rapidly create these types of business applications is an integral part of any digital transformation strategy and helps drive faster innovation, better decisions and exceptional business outcomes.

A key component in building these business applications is the Oracle Integration Cloud Process Automation capabilities that include:

- Model business processes using a variety of drag and drop low code tools, optimized to the way business leaders and practitioners communicate and think about business operations.
- Integrate with any system of record, including SaaS, on-prem, and legacy enterprise applications, regardless of the underlying technology.
- Design decision models to intelligently conform business processes to governance and compliance policies.
- Create real-time operational analytics models that provide business leaders the insights needed to drive better business decisions.
- One-click deployment to test and production environments.
- Contextually embed Process Apps where you already engage your customers and employees; in internal web sites, customer portals and enterprise apps.

Scenario

This workshop scenario is based on a fictitious company called NexGen that sells and services computer products.

NexGen uses Oracle Service Cloud to manage service incidents, but the process of managing customers, product registrations and warranty claims is a much broader problem than simply managing an incident. Today, this process requires a lot of time consuming manual steps, affords little visibility to the management team and creates a disconnected customer experience.

To better serve their customers, NexGen needs to transform the way they manage product warranties. They've defined the following business objectives:

- Make it easier for customers to register products and submit warranty claims.
- Integrate the warranty claims business process with their incident management system.
- Gain operational insight into both key business indicators and the state of individual warranty claims.

To accomplish these business objectives, NexGen will digitize, automate and streamline the business processes required to manage warranties across their product portfolio. Using Oracle Integration Cloud, NexGen can:

- Design, secure, publicize and manage APIs to provide service partners access to warranty service incidents.
- Integrate Oracle Service Cloud business objects and events for partner and customer facing applications.
- Automate the end to end warranty claim business processes to drive faster innovation, better decisions and exceptional business outcomes.
- Provide real-time Insights into business operations.



Lab Overview

If you have not already done so, we encourage you to complete the companion training sessions on Oracle Integration Cloud API Management, Integration and Insight. In these training sessions, you will have created the API, integration and analytic model digital assets for the warranty claim use case. In this lab you will leverage those assets to build the business applications to digitize and automate the end to end warranty claim business process.

The lab is divided into four sections, each with one or more objectives.

In **Section 1** you will review and run a fully functioning Warranty Claim Process App. This will introduce you to the structure and artifacts that make up a Process App and how to run a Process App from the native Process Workspace.

In **Section 2** you will see how Process Apps can be surfaced outside of the Process Workspace. You will create a simple customer facing portal to surface the Warranty Claim Process App. Run the application again to see how easy it is to use Process Apps for stylized contextual user experiences.

In **Section 3** you will get into the details of Process App composition. You will build a new Process App to allow customers to register their purchased products that they can later submit warranty claims against.

Section 4 will tie everything together and allow you to run the full warranty business process lifecycle from registering yourself and the products you've purchase, to making warranty claims when you have an issue with those products, to working the resulting Service Cloud incident.

Get ready to learn about:

- The key components that make up a Process App
- Running a Process App in Process Workspace
- Hosting a Process App in a customer facing website
- Creating and configuring a Process QuickStart App
- Deploying a Process App
- Working with Process App Web Forms
- Creating a business process model
- Connecting a Process App to enterprise applications
- Analyzing real-time business Insights



Pre-Requisites

You will only need a web browser to complete this lab. We recommend using Google Chrome.

You have been provided access to three separate environments to complete this lab.

- Oracle Integration Cloud
- Oracle Content and Experience Cloud
- Oracle Service Cloud (RightNow)

Your lab coordinator will have provided you the URL, username and password for each of these systems.



Section 1: A Warranty Claim Application

In this section you will review and run a fully functioning Warranty Claim Process App. This will introduce you to the structure and artifacts that make up a Process App and how to run a Process App from the native Process Workspace.

Section Objectives

This section contains the following objectives:

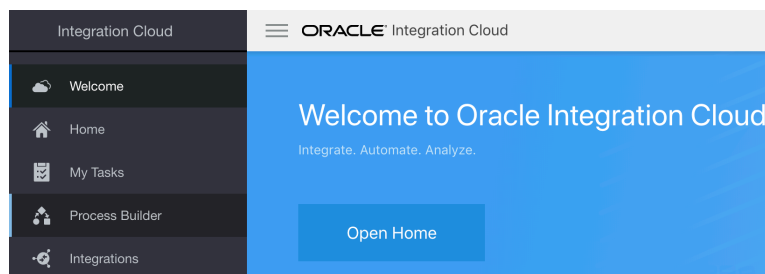
- Examine the components that make up the Warranty Claim Process App
- Run the Warranty Claim Process App from Process Workspace

Examine a Process App

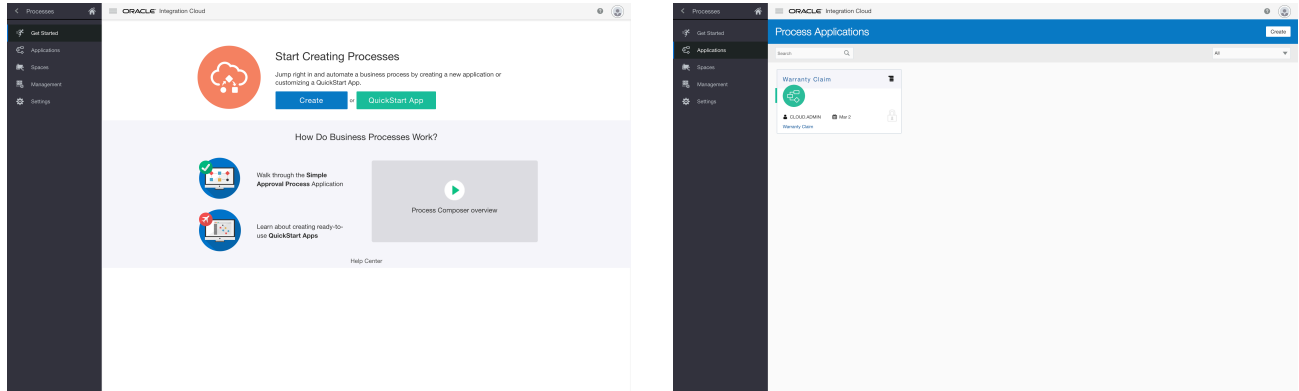
A Process App is composed of a number of different artifacts, including process flows, web forms, decision models, and connections to the outside world. You have been provided Viewer access to a Warranty Claim Process App. Use this objective to explore the application artifacts. Take a look at the web form and process flow to get an idea of what the application does.

Expected Completion Time: 5 minutes

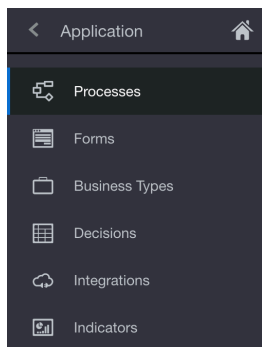
1. Login to Oracle Integration Cloud using the URL and credentials provided for your lab session.
2. Open **Process Builder**. Process Builder is where you compose Process Apps. In this objective you will be reviewing an existing application.



3. Open the **Warranty Claim** Process App. If this is the first time your userId has been used in your lab environment, it is possible you will see the **Start Creating Processes** page. If so, simply click **Applications**.



4. Review the Warranty Claim **Processes**, **Forms** and **Integrations**.

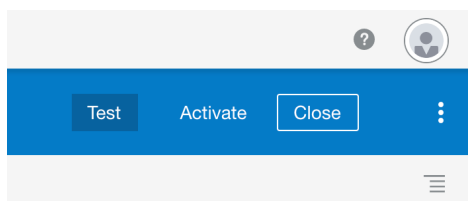


Run a Process App from Workspace

The Process Workspace provides a single pane of glass to start, run, track and manage Process Apps. The Warranty Claim Process App has already been deployed for you. Run the application by submitting a Warranty Claim request.

Expected Completion Time: 5 minutes

1. Starting from the opened Warranty Claim application, Click the **Test** button. The **Test Application** page is where you can deploy the application to the test partition and open the Process Workspace in **Testing Mode**. The application has already been deployed for you.



- Click **Try in Test Mode** to open the Process Workspace.

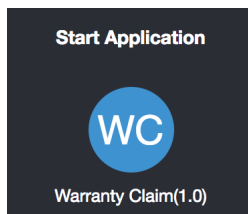
Last active Application Play Try in Test Mode

Active Revision	Activation Date	Status
Warranty_Claim	3/2/2018	On

- Note the Process Workspace is opened in **Testing Mode** that restricts access to those applications that have been deployed to the test partition. Switching **Testing Mode Off** restricts access to those applications that have been deployed to the production partition. Keep **Testing Mode On**.

✕ Testing Mode • On • Off

- Start the Warranty Claim application.



- The Warranty Claim form demonstrates a dynamic web form connected to Service Cloud through the Process App Integration artifacts to provide real-time data. If the email address entered has not previously been registered with Service Cloud, the user will not be able to submit a warranty claim. Complete the form and click **Submit**.

- Enter **nathan.angstadt@oracle.com** in the **Email** field.
- Select a product of your choice from the **Registered Products** field.
- Select an issue of your choice from the **Issue** field.
- Enter a description of your choice in the **Description** field.

✕ Testing Mode • On • Off

Start Application for Warranty Claim(1.0) Submit Save Discard

Please enter your email.

Warranty Claim Number
71462da1-cc72-48c6-626b-2a3151081df7

Email *

nathan.angstadt@oracle.com

Welcome back Nathan Demo. Please select your product.

Registered Product *

CYBER Gamer Xtreme IV | 2018-02-28 | A520532

Issue *

Battery does not charge

Description



Section 2: Warranty Claim Application User Experience

In this section you will see how Process Apps can be surfaced outside of the Process Workspace. You will create a simple customer facing portal to surface the Warranty Claim Process App. Run the application again to see how easy it is to use Process Apps for stylized contextual user experiences.

Section Objectives

This section contains the following objectives:

- Embed the Warranty Claim Process App in a customer portal website
- Run the embedded Warranty Claim Process App from the customer portal website

Embed a Process App in a Website

The Process App user experience components are delivered as reusable UI components that can be embedded in and stylized for most applications and branded web sites. Build a customer portal to host the Warranty Claim Process App where NexGen customers can come to submit warranty claims.

You will be using Content and Experience Cloud to build the customer portal. Content and Experience Cloud provides out of the box components to drag and drop Process App UI components to a site page. Process App UI components can easily be embedded in other applications and web sites using simple customizations based on the underlying application technology.

Expected Completion Time: 5 minutes

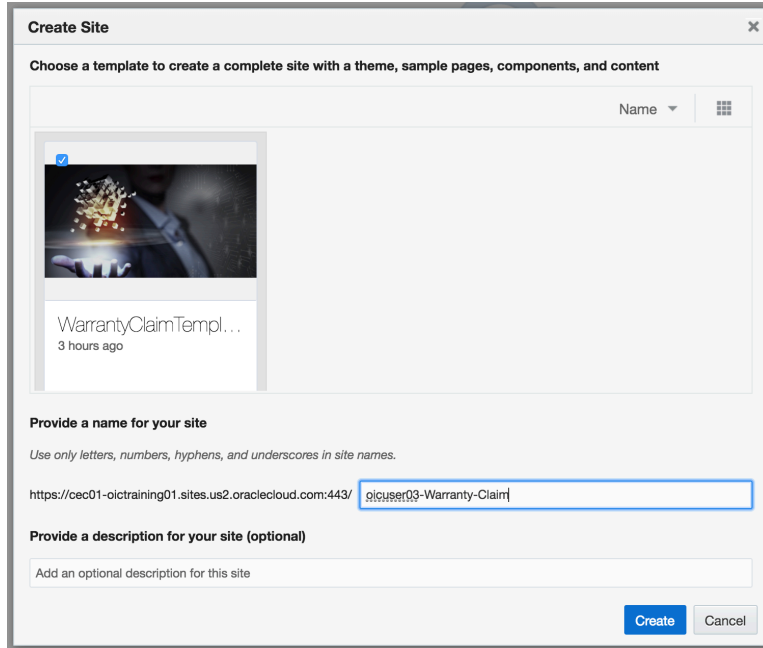
1. Login to Oracle Content and Experience Cloud using the URL and credentials provided for your lab session.

Note: In a production environment, Oracle Integration Cloud and Content and Experience Cloud would share identities. Since this is a lab environment, you need to ensure you keep your Oracle Integration Cloud browser open, so please open Oracle Content and Experience Cloud in a new browser tab.

2. Create a new site by clicking the **Create** button. Complete the **Create Site** form and click **Create**.

- Select the **Warranty Claim Template**.
- Enter a site name of your choice in **Provide a name for your site**.

Note: You are sharing the Oracle Content and Experience Cloud environment with other lab participants. You will therefore need to ensure your Warranty Claim site has a unique name. We recommend appending your username to your site name (e.g. oicuser03-Warranty-Claim).



Create Site

Choose a template to create a complete site with a theme, sample pages, components, and content

Name ▾

WarrantyClaimTempl...
3 hours ago

Provide a name for your site

Use only letters, numbers, hyphens, and underscores in site names.

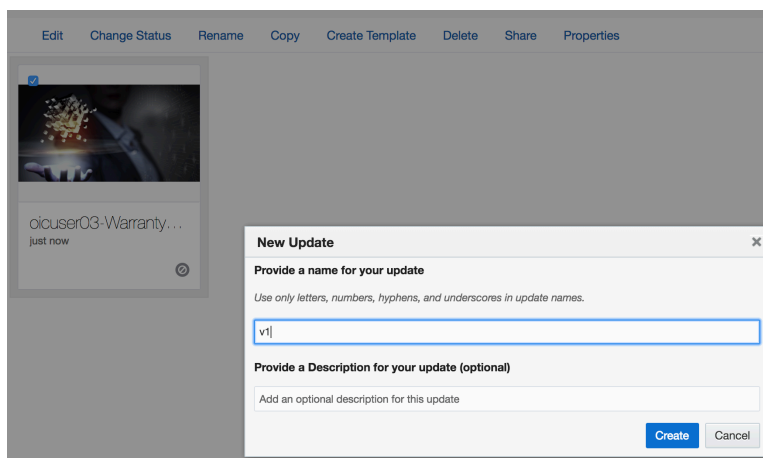
https://cec01-oictraining01.sites.us2.oraclecloud.com:443/ oicuser03-Warranty-Claim

Provide a description for your site (optional)

Add an optional description for this site

Create Cancel

3. Edit your site by clicking the **Edit** action. Enter a name for your update (e.g. v1) in the **New Update** window. Click **Create**.



Edit Change Status Rename Copy Create Template Delete Share Properties

oicuser03-Warranty...
just now

New Update

Provide a name for your update

Use only letters, numbers, hyphens, and underscores in update names.

v1

Provide a Description for your update (optional)

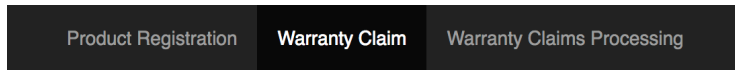
Add an optional description for this update

Create Cancel

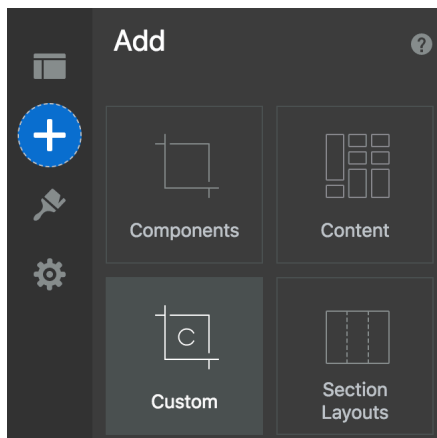
4. Change the mode from View to **Edit**.



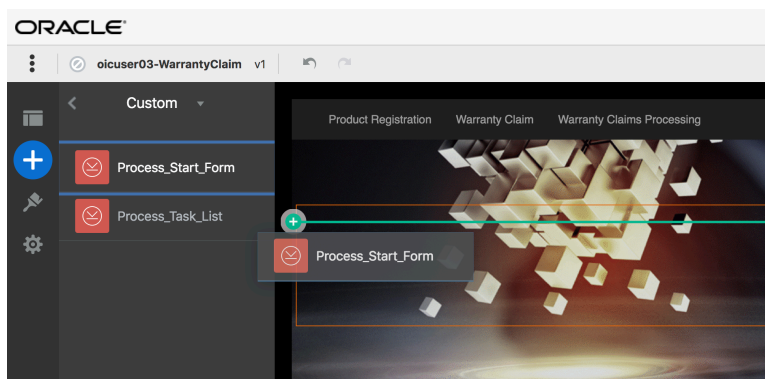
5. The site template you are using comes pre-built with three pages; one for the customer to register products (we will use this page later in the lab), one for the customer to make warranty claims and one for the NexGen warranty claims process team to work assigned tasks. Then Select the **Warranty Claim** page.



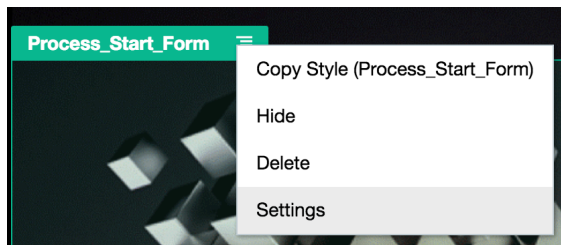
6. Select the **Custom** palette group.



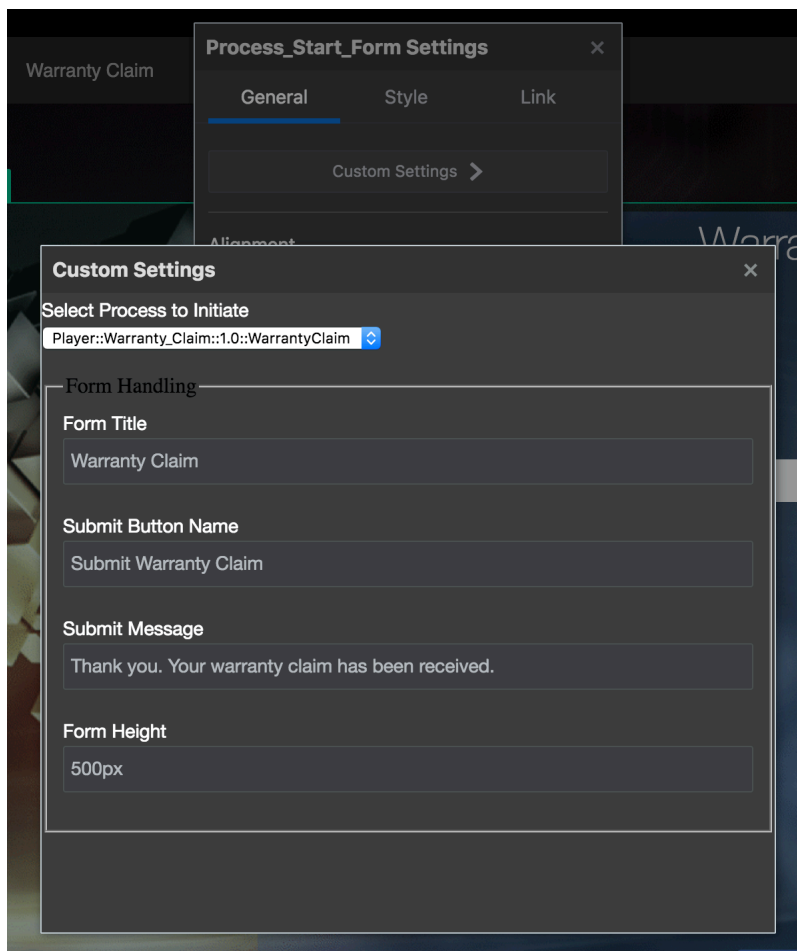
7. Drag and drop the **Process Start Form** component to the page.



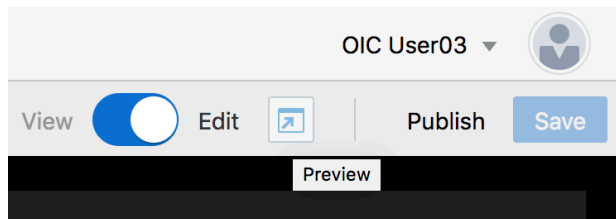
8. Open the component **Settings**.



9. Complete the **Custom Settings** form and close (X) both the **Custom Settings** and **Process Start Form Settings** page.
 - Select **Player::Warranty_Claim::1.0:WarrantyClaim** from **Select Process to Initiate**.
 - Enter a **Form Title** of your choice.
 - Enter a **Submit Button Name** of your choice.
 - Enter a **Submit Message** of your choice.



10. Save the page by clicking **Save** and open in **Preview**.



Run an Embedded Process App from a Web Site

Process Workspace is one place to run applications. A much more compelling experience is to embed the applications where you already engage your customers and employees. This allows application users to remain in their current context without the need to switch to a completely different experience (e.g. Process Workspace). Submit a Warranty Claim request. Notice the difference in the Process Workspace and embedded user experience.

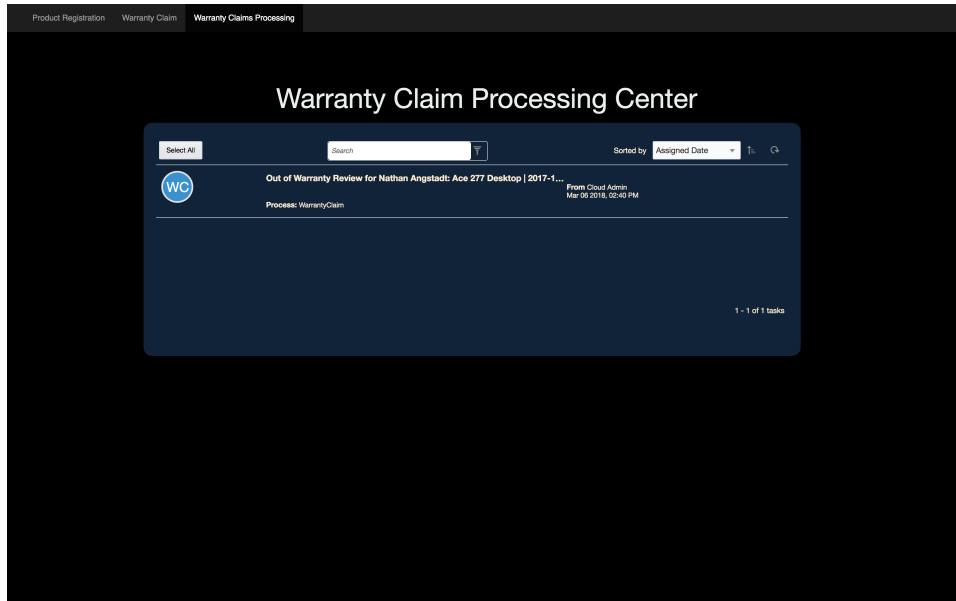
Expected Completion Time: 5 minutes

1. Select the Warranty Claim page. Complete the form and click your customized **Submit** button.

- Enter **nathan.angstadt@oracle.com** in the **Email** field.
- Select a product of your choice from the **Registered Products** field.
- Select an issue of your choice from the **Issue** field.
- Enter a description of your choice in the **Description** field.

A screenshot of a web application interface. At the top, there are three tabs: 'Product Registration', 'Warranty Claim' (which is active), and 'Warranty Claims Processing'. The main content area has a dark background with a glowing cube graphic on the left. On the right, there is a 'Warranty Claim' form. The form has a header 'Warranty Claim' and a 'Warranty Claim Number' field with the value 'ba267808-0a82-0e75-07a3-ed901b0b72e6'. Below this is an 'Email' field with the value 'nathan.angstadt@oracle.com'. A message says 'Welcome back Nathan Angstadt. Please select your product.' Below this is a 'Registered Product' dropdown menu with the value 'Ape 277 Desktop | 2017-11-18 | 1234'. Below that is an 'Issue' dropdown menu with the value 'Battery does not charge'. Below that is a 'Description' text area with the value 'Battery is not working.'. At the bottom right of the form is a blue button labeled 'Submit Warranty Claim'.

2. Process App tasks can also be embedded in web sites or enterprise applications. To simplify the lab, we are hosting the Warranty Claim Processing Center tasks in the same customer portal site. In a real production solution, these tasks would be hosted in a separate web site or other enterprise application. Select the **Warranty Claims Processing** page. Depending on the registered products you chose, you may be assigned an **Out of Warranty** task. Complete this task by clicking the task, review the information and click **Approve**. You will also be assigned an **Incident Created Notification** task. Click the Refresh button to refresh your tasks and complete the task.





Section 3: Building a Product Registration Application

In this section you will get into the details of creating a Process App. You will build a new application to allow customers to register their purchased products that they can later make warranty claims against. The purpose of this application is not to provide implement an exhaustive business case, but to demonstrate the principles of how to create a Process App.

Section Objectives

This section contains the following objectives:

- Create a Product Registration Process App from QuickStart. The QuickStart gives you a starting point for the process web form and process flow.
- Import the Service Cloud Integrations needed by the Product Registration Process App.
 - **Query Contact By Email:** This integration queries Service Cloud for the existence of a Contact with a given email. The Product Registration Form will use this integration to determine whether the customer registering a product is a new customer or a returning customer. Based on the results of this query, the customer will be prompted to provide basic contact information, such as first and last name, or just enter their product information.
 - **Get Products:** This integration queries Service Cloud for a list of products that can be registered for a warranty. The Product Registration Form will use this integration to prompt the user for which product they are registering.
 - **Create Contact:** This integration creates a Contact in Service Cloud that represents a customer who has registered products.
 - **Create Asset:** This integration creates an Asset in Service Cloud that represents a product registered by a customer.
- Update the Product Request Web Form. The web form has been started for you as a starting point but you will need to update and add new UI components to complete the web form.
 - Update the **Email** UI component to query for a Service Cloud Contact based on the entered email address.
 - Add a Message to prompt an existing Service Cloud Contact to register their product. This message should only be shown if the email address is associated with an existing Service Cloud Contact.
 - Add a Message to prompt new users to enter their contact details and register their product. This message should only be shown if the email address is not associated with an existing Service Cloud Contact.
 - Add UI components to collect first name and last name. Hide, show and change the required property of these fields based on whether the user is an existing Service Cloud Contact or a new user.
 - Add a selector to display a list of Service Cloud products. The user will select the products they need to register from this list.
 - Add UI component to collect the product serial number.
 - Add a UI component to collect the product purchase date. (**Note:** the purchase date format must be “yy-MM-dd” to work correctly with the Service Cloud integrations.
 - Update the **Registered Owner Name** UI component to automatically populated with the **contact.lookupName** returned by the Service Cloud Contact query.
 - Update the **Registered Owner Id** UI component to automatically populated with the **contact.id**

returned by the Service Cloud Contact query.

- Add a new Web Form Presentation to review a newly created Service Cloud Asset.
 - Add a UI component to display the Asset Name.
 - Add a UI component to display the Asset Id.
- Complete the Process Flow.
 - Replace the **Create Contact** placeholder activity with the **Create Contact** integration.
 - Replace the **Create Asset** placeholder activity with the **Create Asset** integration.
 - Replace the **Review Product Registration** placeholder with a **Submit** Task activity that shows the Asset Review presentation.
- Deploy the Product Registration Process App to Test.

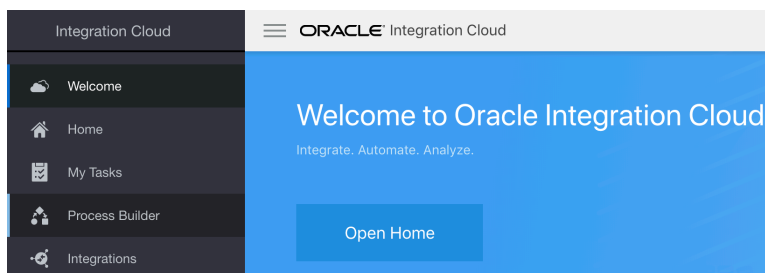
Create a Process App from QuickStart

QuickStart Apps make it easy to rapidly create, personalize, test and deploy fully functioning Process Apps from a catalog of predefined application templates. This empowers the business to take greater ownership in managing their own Process Apps while removing IT as a bottleneck to business innovation. New QuickStart Apps can be published to the catalog, making it easy to extend the library of available Process App templates.

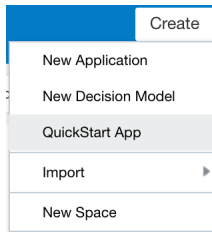
A Product Registration QuickStart App has already been published to the catalog for your use. Create a Product Registration Process App from this QuickStart.

Expected Completion Time: 5 minutes

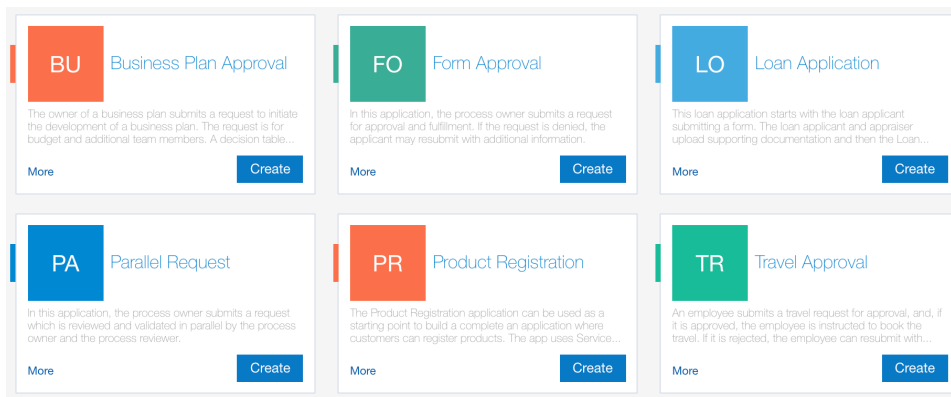
1. Login to Oracle Integration Cloud using the URL and credentials provided for your lab session.
2. Open **Process Builder**.



- Click the **Create** button, then **QuickStart App**.

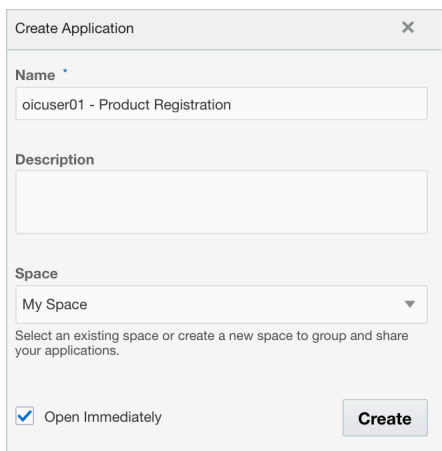


- Click **Create** on the **Product Registration** QuickStart.



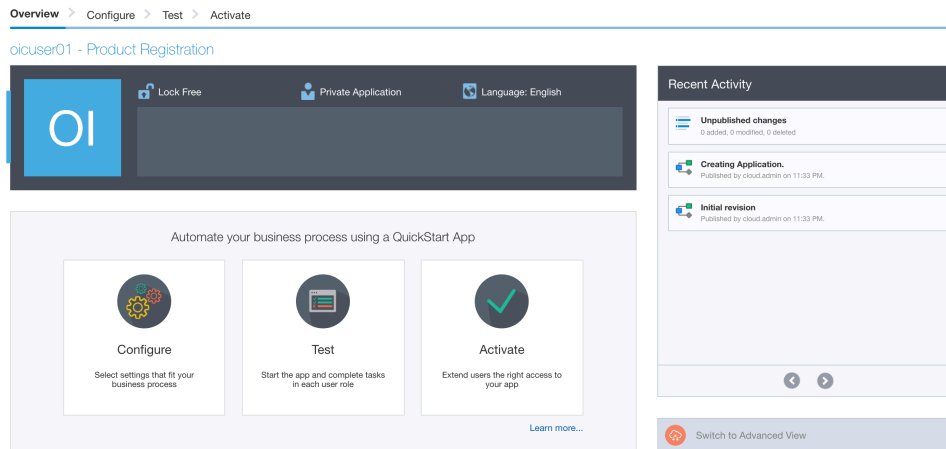
- Complete the **Create Application** form. Leave the **Open Immediately** option checked and click **Create**.
 - Enter an application name in **Name**.

Note: You are sharing the Oracle Integration Cloud environment with other lab participants. You will therefore need to ensure your Product Registration app has a unique name. We recommend appending your username to your Process App name (e.g. oicuser01- Product Registration).



6. A Process App created from a QuickStart opens in **Personalization View**. This view provides a simplified limited experience for configuring and personalizing the application. Clicking **Switch to Advance View** opens the application in the full Process Composer and gives access to all of the underlying Process App artifacts.

Click **Switch to Advanced View** to open the full Process Composer.



7. The QuickStart already contains a process and form named **Warranty Registration**. Review the process and form.

Import Integration Connections

Most Process Apps need to connect to one or more systems of record, both to retrieve data for use in web forms or to create and update transactions in the process flow.

The Product Registration Process App requires four integrations that interact with Service Cloud:

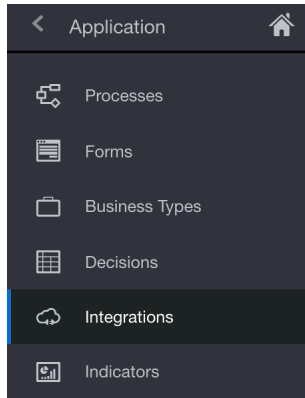
- **Query Contact By Email:** This integration queries Service Cloud for the existence of a Contact with a given email. The Product Registration Form will use this integration to determine whether the customer registering a product is a new customer or a returning customer. Based on the results of this query, the customer will be prompted to provide basic contact information, such as first and last name, or just enter their product information.
- **Get Products:** This integration queries Service Cloud for a list of products that can be registered for a warranty. The Product Registration Form will use this integration to prompt the user for which product they are registering.
- **Create Contact:** This integration creates a Contact in Service Cloud that represents a customer who has registered products.
- **Create Asset:** This integration creates an Asset in Service Cloud that represents a product registered by a customer.

Oracle Integration Cloud is a persona-based platform that allows teams with different skillsets and specializations to work together to deliver solutions that transform the business. For example, integrations may be created by teams who have deep insight into the corporate systems of record and skillsets necessary to understand how to best organize orchestrating interactions between these systems. Business applications that govern the business process may be created by teams closer to business operations, individuals who have a deep understanding of the overall business objectives and customer engagement process. Oracle Integration Cloud allows these teams to work together, each leveraging their areas of expertise, each contributing to the technical and business objectives. Connecting a Process App to a system of record is as easy as importing integration assets into the Process App project. Individuals building Process Apps need not have any understanding of the technical details behind connecting to these systems.

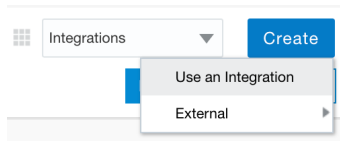
In this objective you will import integrations created for the Product Registration Process App.

Expected Completion Time: 5 minutes

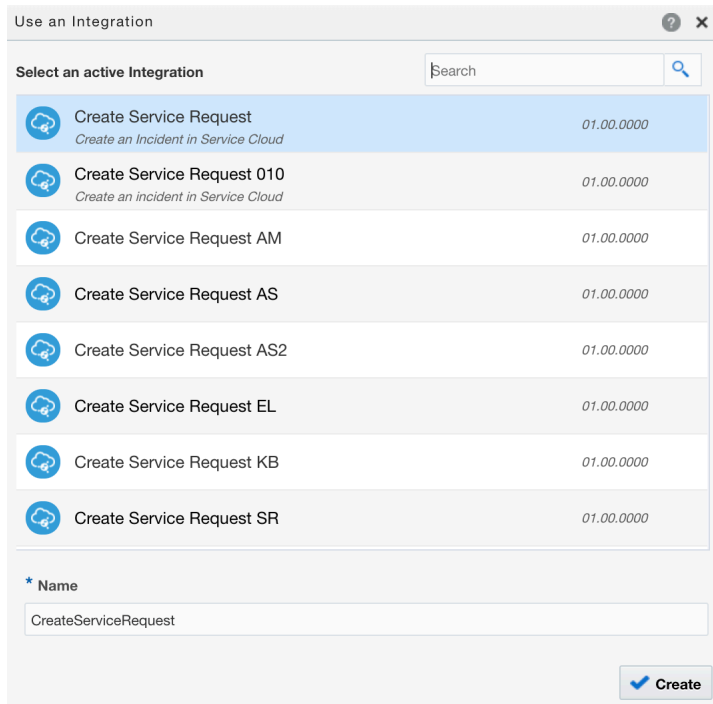
1. Click Integrations to open the Integrations artifacts page.



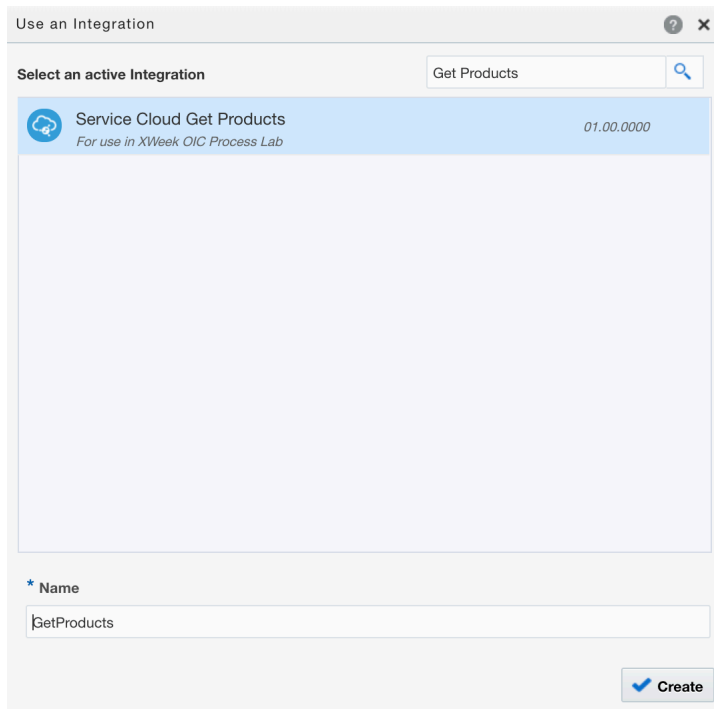
2. Click on **Browse Integrations** or **Create**, then **Use an Integration**.



3. A list of the available integrations is displayed.

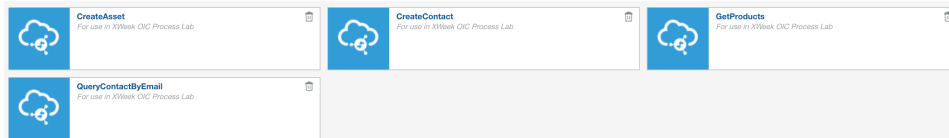


4. For each of the integrations below, search for the integration, change the **Name** and click **Create**. The table below specifies the integrations and names to create.



Integration	Name
Service Cloud Get Products	GetProducts
Service Cloud Query Contact By Email	QueryContactByEmail
Service Cloud Create Contact	CreateContact
Service Cloud Create Asset	CreateAsset

5. You should now have four integration assets in your Process App project.



Build a Web Form

Web Forms allow for the quick and easy development of forms to be used by your Process App. The forms builder allows you to easily drag and drop UI components such as Input Text, Dates, Number, Email, Image and Table components onto the form.

You can leverage Integration Connectors to populate data on your form.

As fields are dropped onto the form, the underlying data model is automatically created for you and can be leveraged in the process.

Guided and contextual no-code rules can be created to define form business logic. Presentations can be created that use the same data model, but allow for different views on that data.

Preview Mode can be used to test your forms to see how it will look in different devices

Using Global Connectors

Integration connectors can be used to provide real-time access to systems of record. Connectors can be defined locally for individual UI components or for global use across the entire form.

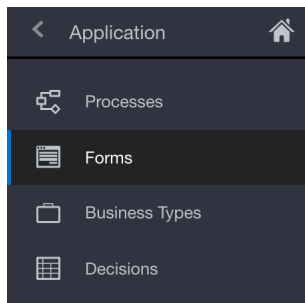
Create a global connector for your web form.

Expected Completion Time: 5 minutes

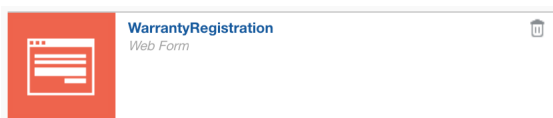
These instructions have been supplemented with a video that details the steps for completing the objective.

Open the video folder and watch the video named: 1. Creating a Web Form Global Connector.

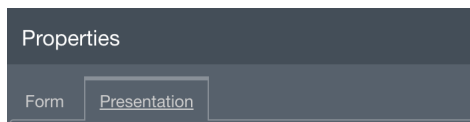
1. Click **Forms** to open the Forms artifacts page.



2. Click the **WarrantyRegistration** form to open the form.



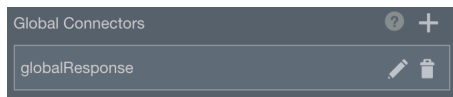
3. Click **Presentation** in the **Properties** window.



4. Click the **+** button to create a new **Global Connector**.



- Click the pencil button to edit the new Global Connector.



- Configure the connector.

Which connector do you want to call?

Connector: QueryContactsByEmail

Resource: resources

Operation: getResource

Skip Upon Load: Yes

Template Parameters:

Control Value → Email

Where do you want to store the response data?

Response: queryContactResponse

A screenshot of the connector configuration form. At the top, there is a green button labeled 'Connector Call'. Below it, the form is divided into sections. The first section, 'Which connector do you want to call?', contains three dropdown menus: 'Connector' (set to 'QueryContactsByEmail'), 'Resource' (set to 'resources'), and 'Operation' (set to 'getResource'). Below these are two checkboxes: 'Skip Upon Load' (checked) and 'Don't show loading spinner' (unchecked). The second section, 'Template Parameters', has a label 'email' and a dropdown menu 'Control value' (set to 'Email'). The third section, 'Where do you want to store the response data?', has a text input field containing 'queryContactResponse'.

Main Presentation

The main presentation will represent the form view for the customer to register a product.

Update the form with the additional UI components and business logic needed for the application.

Your completed form should look like this.

Please enter your email.

Email *

Existing Registration Message is computed based on Contact Query results

It looks like this is your first time here. Please enter your name and new product information.

First Name Last Name

Product * Serial Number * Purchase Date *

Registered Owner Name Registered Owner Id

Expected Completion Time: 30 minutes

These instructions have been supplemented with a video that details the steps for completing the objective.

Open the video folder and watch the video named: 2. Updating the Main Presentation.

1. Modify the **Email** component named **Email** with the following properties:

Name: Email (this is already set)

Label: Email (this is already set)

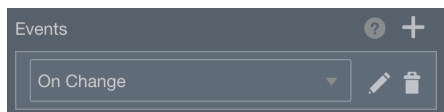
Binding: email (this is already set)

Required: Yes

OnChange Event:

Action:

Presentation → Refresh Global Connector → queryContactResponse



Action		
Control Name	Action	Connector
Presentation	Refresh Global Connector	queryContactResponse

2. Add a new **Message** component with the following properties:

Name: ExistingRegistrationMessage

Binding: existingRegistrationMessage

Default Text: Existing Registration Message is computed based on Contact Query results

Type: Heading 3

Hide: Yes

Computed Value:

*Function → Concat → Constant → **Welcome back** → Connector Data → `queryContactResponse.contact.lookupName` → Constant → **. Please register your new product.***

The screenshot shows the 'Computed Value' editor in Oracle APEX. It features a dark header with a checked 'Computed Value' checkbox, a help icon, and an 'Edit' button. Below the header, the expression is built using a sequence of components: a 'Function' dropdown set to 'Concat', followed by three arguments in parentheses. The first argument is a 'Constant' with the value 'Welcome back'. The second argument is a 'Connector Data' component with the value 'queryContactResponse.contact.lookupName'. The third argument is another 'Constant' with the value '. Please register your new product.' Each argument is shown in its own box with a type dropdown and a value field.

3. Add a new **Message** component with the following properties:

Name: NewRegistrationMessage

Binding: newRegistrationMessage

Default Text: It looks like this is your first time here. Please enter your name and new product information.

Type: Heading 3

Hide: Yes

4. Add a new **Panel** component with the following properties:

Name: NewOwnerPanel

Layout: Horizontal

Hide: Yes

5. Add a new **Input Text** component with the following properties and place inside the **NewOwnerPanel**:

Name: FirstName

Label: First Name

Binding: firstName

6. Add a new **Input Text** component with the following properties and place inside the **NewOwnerPanel**:

Name: LastName

Label: Last Name

Binding: lastName

7. Add a new **Panel** component with the following properties:

Name: ProductPanel

Layout: Horizontal

Hide: Yes

8. Add a new **Select** component with the following properties and place inside the **ProductPanel**:

Name: Product

Label: Product

Binding: product

Options Source: Connector

Connector: GetProducts

Resource: resource

Operation: getResource

Response:

Options List: response.items

Label Binding: lookupName

Value Binding: Id

The screenshot shows the configuration interface for a Select component in Oracle APEX. It features a dark-themed background with various input fields and dropdown menus. At the top, there's a section for 'Options Source' with three radio buttons: 'Static', 'From Data', and 'Connector', where 'Connector' is selected. Below this are three dropdown menus for 'Connector' (set to 'GetProducts'), 'Resource' (set to 'resources'), and 'Operation' (set to 'getResource'). A dashed horizontal line separates the source configuration from the response configuration. Under 'Response', there are three text input fields: 'Options List' (containing 'response.items'), 'Label Binding' (containing 'lookupName'), and 'Value Binding' (containing 'id'). Each input field has a small question mark icon to its right.

Required: Yes

- 
9. Add a new **Input Text** component with the following properties and place inside the **ProductPanel**:

Name: SerialNumber

Label: Serial Number

Binding: serialNumber

Required: Yes

10. Add a new **Date** component with the following properties and place inside the **ProductPanel**:

Name: PurchaseDate

Label: Purchase Date

Binding: purchaseDate

Required: Yes

Format: yy-MM-dd (**Note:** the purchase date format must be “yy-MM-dd” to work correctly with the Service Cloud integrations)

11. Modify the **Input Text** component named **RegisteredOwnerName** with the following properties:

Name: RegisteredOwnerName (this is already set)

Label: Registered Owner Name (this is already set)

Binding: contactName (this is already set)

Computed Value:

Connector Data -> **queryContactResponse.contact.lookupName**

A dark grey panel with a blue checkmark icon and the text "Computed Value" on the left, a question mark icon in the center, and an "Edit" button on the right.

Type	Value
Connector Data	queryContactResponse.contact.lookupName

Hide: Yes

OnChange Event

If: Control → RegisteredOwnerName → Value → Is True

Then:

Action: ExistingRegistrationMessage → Show

Action: NewRegistrationMessage → Hide

Action: NewOwnerPanel → Hide

Action: FirstName → Optional

Action: LastName → Optional

Else:

Else Action: ExistingRegistrationMessage → Hide

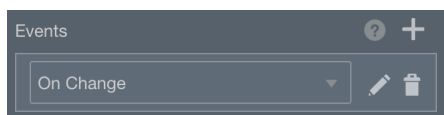
Else Action: NewRegistrationMessage → Show

Else Action: NewOwnerPanel → Show

Else Action: FirstName → Required

Else Action: LastName → Required

Action: ProductPanel → Show

A dark grey panel with the title "Events" and a plus icon on the right. Below is a list with one item "On Change" which has a dropdown arrow, an edit icon, and a delete icon.

If

+ Condition

Type

Control

Control Name

RegisteredOwnerName

Property

Value

is True

Then

+ Action

Action

Control Name

ExistingRegistrationMessage

Action

Show

Action

Control Name

NewRegistrationMessage

Action

Hide

Action

Control Name

NewOwnerPanel

Action

Hide

Action

Control Name

FirstName

Action

Optional

Action

Control Name

LastName

Action

Optional

Else

+ Else If

+ Else Action

Action

Control Name

ExistingRegistrationMessage

Action

Hide

Action

Control Name

NewRegistrationMessage

Action

Show

Action

Control Name

NewOwnerPanel

Action

Show

Action

Control Name

FirstName

Action

Required

Action

Control Name

LastName

Action

Required

Action

Control Name

ProductPanel

Action

Show

12. Modify the **Number** component named **RegisteredOwnerId** with the following properties:

Name: RegisteredOwnerId (this is already set)

Label: Registered Owner Id (this is already set)

Binding: contactId (this is already set)

Computed Value:

Connector Data -> **queryContactResponse.contact.id**

The screenshot shows a configuration panel for a 'Computed Value'. It has a checked 'Computed Value' checkbox, a question mark icon, and an 'Edit' button. Below this, there are two input fields: 'Type' and 'Value'. The 'Type' field is a dropdown menu currently showing 'Connector Data'. The 'Value' field is a text input containing the expression 'queryContactResponse.contact.id'.

13. Preview Mode can be used to test your forms to see how it will look in different devices. Click **Save** to save your work. Then click **Preview** to review and test your form.

This block contains three buttons: a blue 'Close' button, a white 'Save' button with a blue border, and a grey 'Preview' button with a magnifying glass icon.

Asset Review Presentation

The asset review presentation will represent the form view for the internal Warranty Claim Processing team to review when a new product has been registered and new asset in service cloud has been created.

Your completed form should look like this.

The screenshot shows a form with two sections. The first section is titled 'Asset Name' and has a text input field. The second section is titled 'Asset Id' and has a text input field containing the value '0'. Each section has a vertical ellipsis icon to its left.

Expected Completion Time: 5 minutes

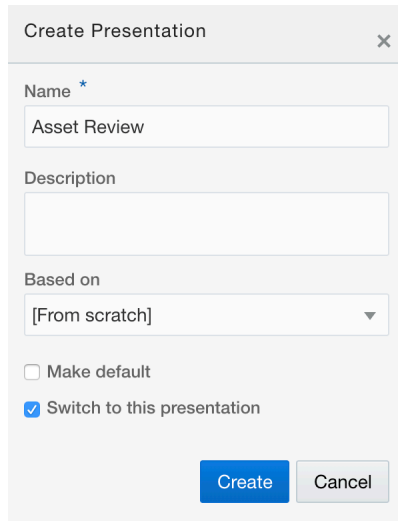
These instructions have been supplemented with a video that details the steps for completing the objective.

Open the video folder and watch the video named: 3. Creating the Asset Review Presentation.

1. Create a new presentation called **Asset Review**.

Based on: [From Scratch]

The screenshot shows a list of presentations. The first item is 'Presentations' with a question mark icon and a plus sign. The second item is 'Main' with a star icon.

A screenshot of a 'Create Presentation' dialog box. It has a title bar with a close button. The form contains a 'Name' field with a red asterisk, containing the text 'Asset Review'. Below it is a 'Description' field. Then a 'Based on' dropdown menu showing '[From scratch]'. At the bottom, there are two checkboxes: 'Make default' (unchecked) and 'Switch to this presentation' (checked). At the very bottom are 'Create' and 'Cancel' buttons.

Create Presentation

Name *

Asset Review

Description

Based on

[From scratch]

☐ Make default

☒ Switch to this presentation

Create Cancel

2. Add a new **Input Text** component with the following properties:

Name: AssetName

Label: Asset Name

Binding: assetName

3. Add a new **Number** component with the following properties:

Name: AssetId

Label: Asset Id

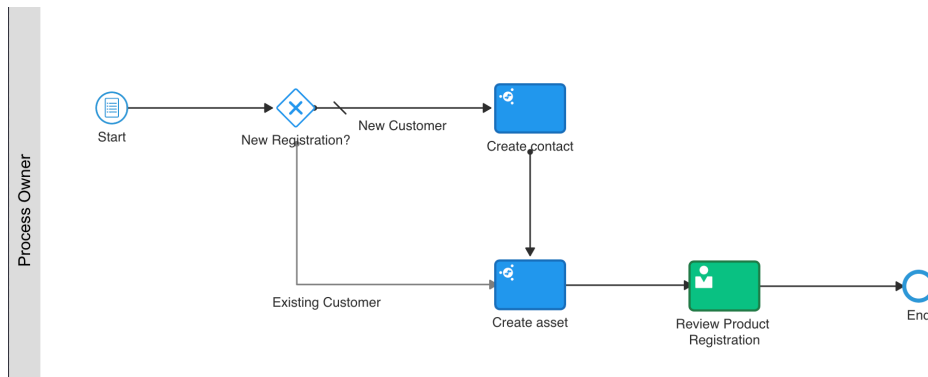
Binding: assetId

Build a Process Flow

Process flows allow you to orchestrate the information, systems, and people required to complete the business process.

The process flow has been seeded with some abstract placeholder activities to get you started on completing the process. You will replace these with executable activities.

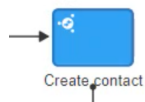
Your completed process flow should look like this.



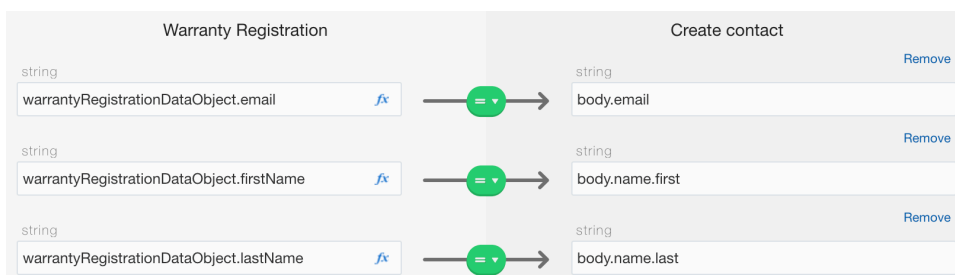
Expected Completion Time: 10 minutes

These instructions have been supplemented with a video that details the steps for completing the objective. Open the video folder and watch the video named: 4. Updating the Process Flow.

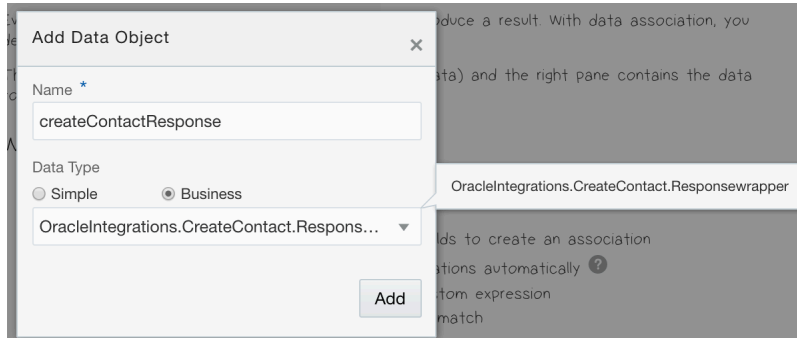
1. Open the **Warranty Registration** Process.
2. Replace the **Create Contact** placeholder activity with the **Create Contact** Integration activity.



3. Complete the **Create Contact** Input Data Association to map the web form data to the integration.



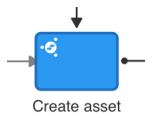
4. Create a Data Object name **createContactResponse** of type **OracleIntegration.CreateContact.Responsewrapper**.



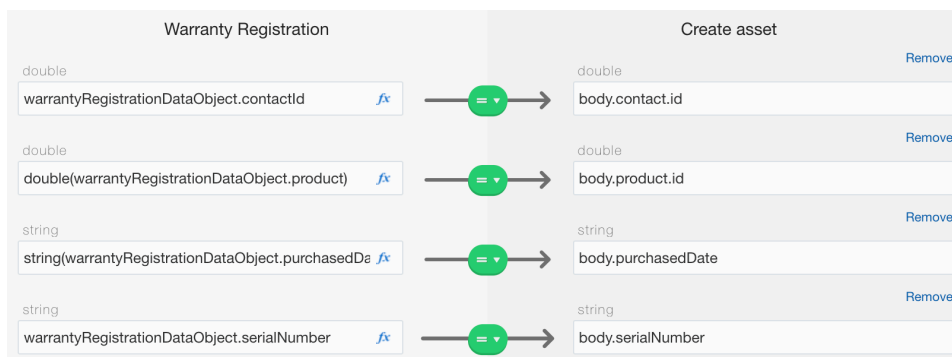
5. Complete the **Create Contact** Output Data Association to map the integration response to the web form data.



6. Replace the **Create Asset** placeholder activity with the **Create Asset** Integration activity.



7. Complete the **Create Asset** Input Data Association to map the web form data to the integration.



8. Create a Data Object name **createAssetResponse** of type **OracleIntegration.CreateAsset.Responsewrapper**.

Add Data Object

Name *

createAssetResponse

Data Type

☐ Simple ☒ Business

OracleIntegrations.CreateAsset.Response...

Add

9. Complete the **Create Asset** Output Data Association to map the integration response to the web form data



10. Replace the **Review Product Registration** placeholder activity with a Human **Submit** Task activity.



11. Define the **Submit** task attributes.

- **Title:** Review Product Registration
- **Form:** WarrantyRegistration
- **Presentation:** Asset Review

The screenshot shows a task configuration interface. On the left, there are sections for 'Who is the assignee?' with a dropdown for 'Any member of current swimlane role', 'How to display the task information?' with a dropdown for 'WarrantyRegistration', and 'Presentation' with a dropdown for 'Asset Review'. On the right, under 'How to identify the task?', there are fields for 'Title' (Review Product Registration), 'Task Summary' (Brief description of the task), 'Due Date' (What is the due date?), and 'Priority' (Normal).

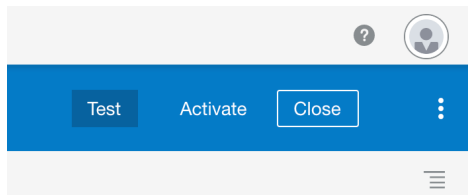
Deploy a Process App for Testing

Process Apps can be deployed to either a training or production partition within a single Oracle Integration Cloud instance. This allows you to easily test your applications within the test partition before making the decision to deploy to production. If multiple Oracle Integration Cloud instances are available, you can deploy your Process Apps to either the local or remote instance, providing complete dev, test and production segregation.

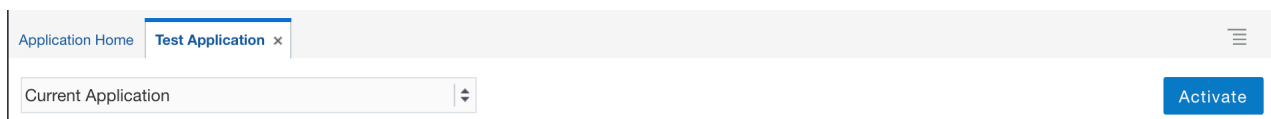
When deploying to the test partition, you have the option to have yourself automatically added to all of the Process App roles. This allows you to immediately run the app end to end without needing to configure roles or login as individual stakeholders. You can of course add additional users to these roles at any time. For this lab, we will deploy to the test partition and take the option to include ourselves in all roles.

Expected Completion Time: 5 minutes

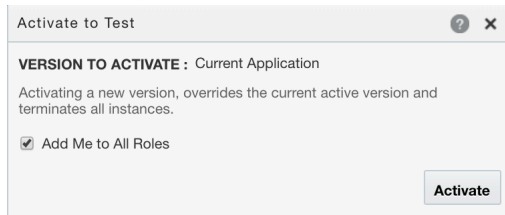
1. Starting from the opened Product Registration application, Click the **Test** button. The **Test Application** page is where you can deploy the application to the test



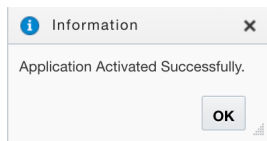
2. Click the **Activate** button. There are actually two Activate buttons; one to deploy to the test partition and one to deploy to the production partition. Ensure you are clicking the **Activate** button in the **Test Application** tab.



3. Keep the **Add Me to All Roles** option checked and click **Activate** in the **Activate to Test** window.



4. You should get a notification that the application was activated successfully.





Section 4: A Complete Warranty Claim Business Process

In this section you will tie everything together and run the full warranty business process lifecycle from registering yourself and the products you've purchased, to making warranty claims when you have an issue with those products, to resolving the resulting Service Cloud incident.

Section Objectives

This section contains the following objectives:

- Add your Product Registration Process App to the Customer Portal
- Register a Product
- Make a Warranty Claim
- Close a Service Cloud Incident
- Gain Insight into the Warranty Claim

Add your Product Registration Application to the Customer Portal

This objective is similar to what you completed in Section 2 when you added the Warranty Claim Process App to your customer portal site. Now you will add your Product Registration Process App to the **Product Registration** site page.

Expected Completion Time: 5 minutes

1. Login to Oracle Content and Experience Cloud using the URL and credentials provided for your lab session.

Note: Remember to keep your Oracle Integration Cloud browser open and open Oracle Content and Experience Cloud in a new browser tab.

2. Open your Warranty Claim site for editing.
3. Select the **Product Registration** page.
4. Select the **Custom** palette group.
5. Drag and drop the **Process Start Form** component to the page.
6. Open the component **Settings**.
7. Complete the **Custom Settings** form and close (X) both the **Custom Settings** and **Process Start Form Settings** page.

- Select **Player::<Your Product Registration Application Name>::1.0:WarrantyRegistration** from **Select Process to Initiate**.
- Enter a **Form Title** of your choice.
- Enter a **Submit Button Name** of your choice.
- Enter a **Submit Message** of your choice.

8. Save the page by clicking **Save** and open in **Preview**.

Register a Product

It's time to run the complete Warranty Claim business process. Start by registering some products. You should expect your form to prompt for your first and last name the first time you register a product. Thereafter, you should be recognized as a returning customer.

Expected Completion Time: 5 minutes

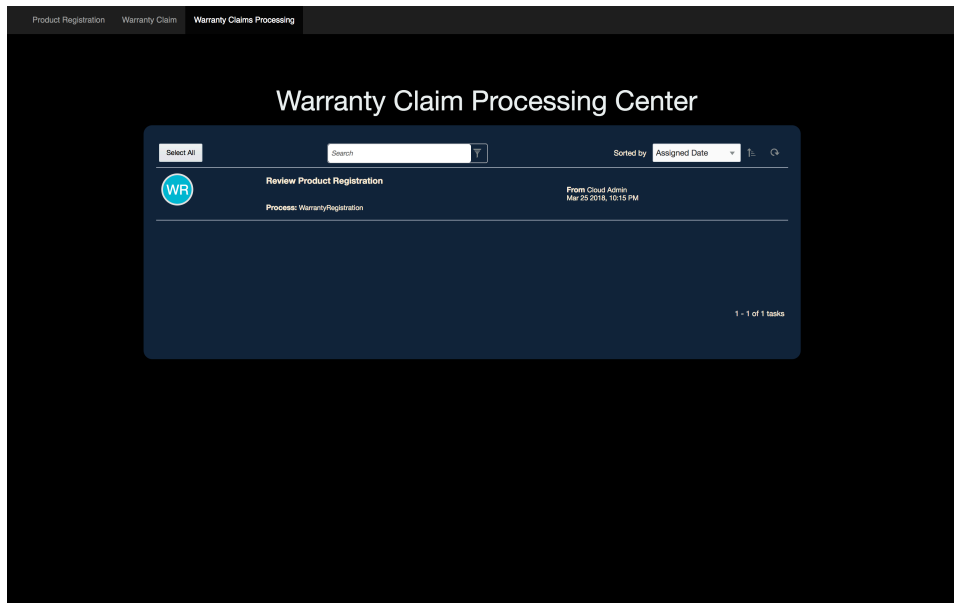
1. Select the **Product Registration** page. Complete the form and click your customized **Submit** button.
 - Enter **<your email address>** in the **Email** field.
 - Enter your first and last name in the **First Name** and **Last Name** fields.
 - Select a product of your choice from the **Product** field.
 - Enter a serial number of your choice in the **Serial Number** field.
 - Enter a date of your **Purchase Date** field. Entering a purchase date older than 90 days from the time you make the warranty claim will require approval.

The screenshot shows a web application interface for 'Product Registration'. At the top, there are three tabs: 'Product Registration' (selected), 'Warranty Claim', and 'Warranty Claims Processing'. The main content area has a dark background with a glowing cube and a hand. A white form overlay contains the following fields:

- Email:** A text input field with the value 'nathan.angstadt@oracle.com'.
- First Name:** A text input field with the value 'Nathan'.
- Last Name:** A text input field with the value 'Angstadt'.
- Product:** A dropdown menu with the value 'Ace 277 Desktop'.
- Serial Number:** A text input field with the value '1234'.
- Purchase Date:** A date picker field with the value '17-11-18'.

Below the form fields is a blue button labeled 'Submit Product Registration'.


2. Select the **Warranty Claims Processing** page. You should expect to be assigned an **Asset Review** task for each product you registered. Complete this task by clicking the task, review the information and click **Submit**.



Make a Warranty Claim

Now it's time to make a warranty claim against some of the products you registered. You should expect to be recognized as a returning customer if you use the same email address that you used to register your products. You should also expect to see your registered products when making the warranty claim.

Expected Completion Time: 5 minutes

1. Select the **Warranty Claim** page. Complete the form and click your customized **Submit** button.
 - Enter **<your email address>** in the **Email** field.
 - Select a product of your choice from the **Registered Products** field.
 - Select an issue of your choice from the **Issue** field.
 - Enter a description of your choice in the **Description** field.
2. Select the **Warranty Claims Processing** page. Depending on the registered products you chose, you may be assigned an **Out of Warranty** task. Complete this task by clicking the task, review the information and click **Approve**. You will also be assigned an **Incident Created Notification** task. Click the Refresh  button to refresh your tasks and complete the task.

Close a Service Cloud Incident

Service Cloud is the foundation for managing incidents. The warranty claim applications and integrations you've built leverage Service Cloud as the underlying system of record.

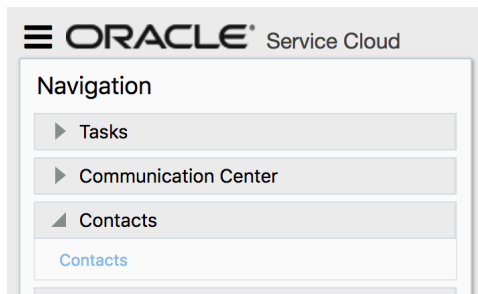
Navigate to Service Cloud to close the incidents opened for the warranty claims you've made.

Expected Completion Time: 5 minutes

1. Login to Oracle Service Cloud using the URL and credentials provided for your lab session.

Note: You will need to be on the Oracle VPN to access Service Cloud.

2. Select Contacts.

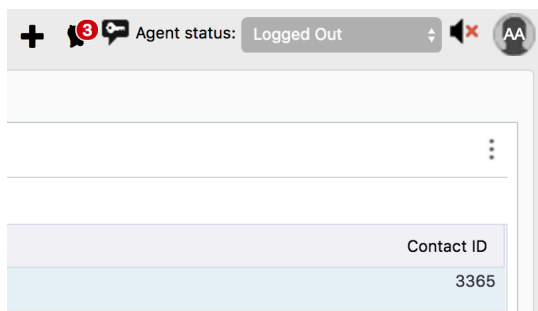


3. Filter by the first and last name you used to register your products.









▲ Filters

Contact ID	Organization ID	Last Name	First Name
<input type="text"/>	<input type="text"/>	<input type="text" value="Angstadt"/>	<input type="text" value="Nathan"/>

4. Double-click you contact id



5. Open each of your incidents.

Contact Incidents			
 Open  Add New  Delete  Print  Copy  Propose			
Status	Source	Reference #	Subject
Unresolved		180301-000040	Keyboard Filthy
Unresolved		180301-000039	Keyboard Filthy

6. Change the Status to Solved and Save and Close the Incident.

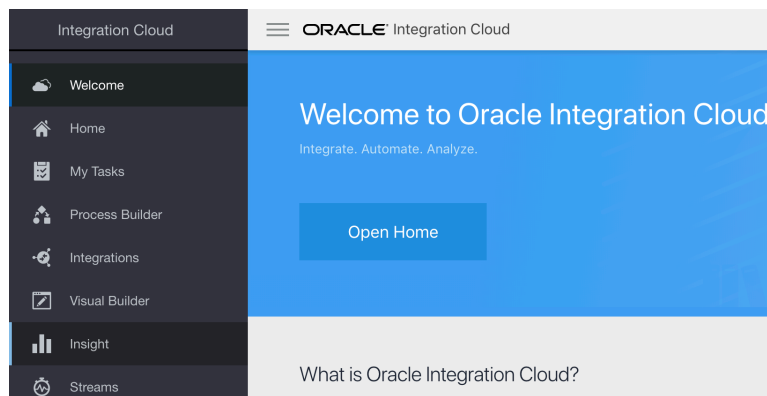
Status *

[No Value]
Solved
Unresolved
Call Back

Gain Insight into the Warranty Claim

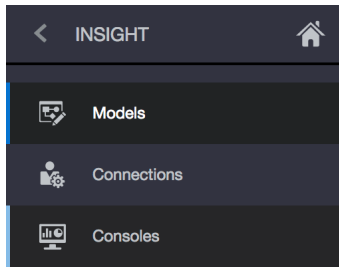
Insight provides the business real-time visibility into business operations both through aggregate operational dashboards and individual warranty claim request milestone progress. Review the progress of your warranty claims.

1. Open **Insight**. Insight provides access to the real-time analytics and milestone progress for your Process Apps and Integrations.



2. Click on **Consoles**.

Note: If you get a **Failed loading the dashboards page** error, please refresh the page.

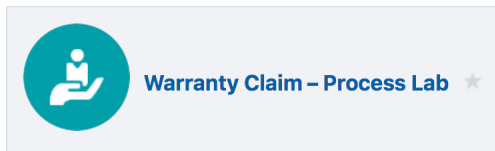


3. Enter **Process** in the Search box.

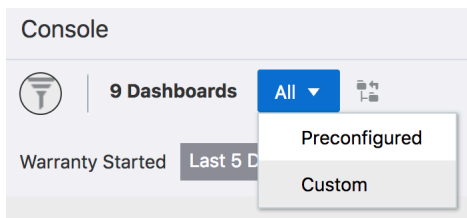
Note: You may see a number of Warranty Claim Consoles that were used as part of the Insight lab. Please ensure you find the Console named **Warranty Claim - Process Lab**.



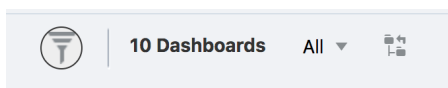
4. Open the **Warranty Claim – Process Lab** Console.



5. Click on **Custom** to display the custom dashboards.



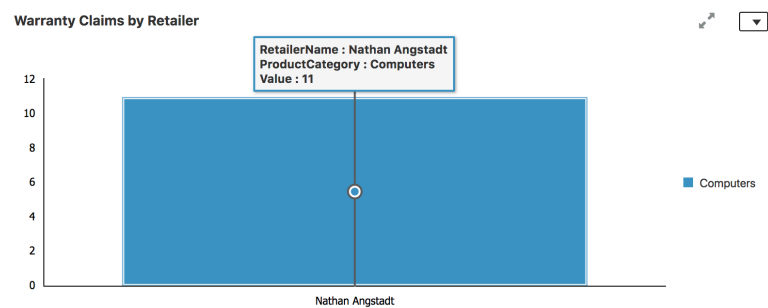
6. Click the Filter.



7. Enter your name in the **Retailer Name** filter and click **Apply**. This is the first and last name you used to register your products.

Warranty Request Outcome
Retailer Name
Customer Satisfaction

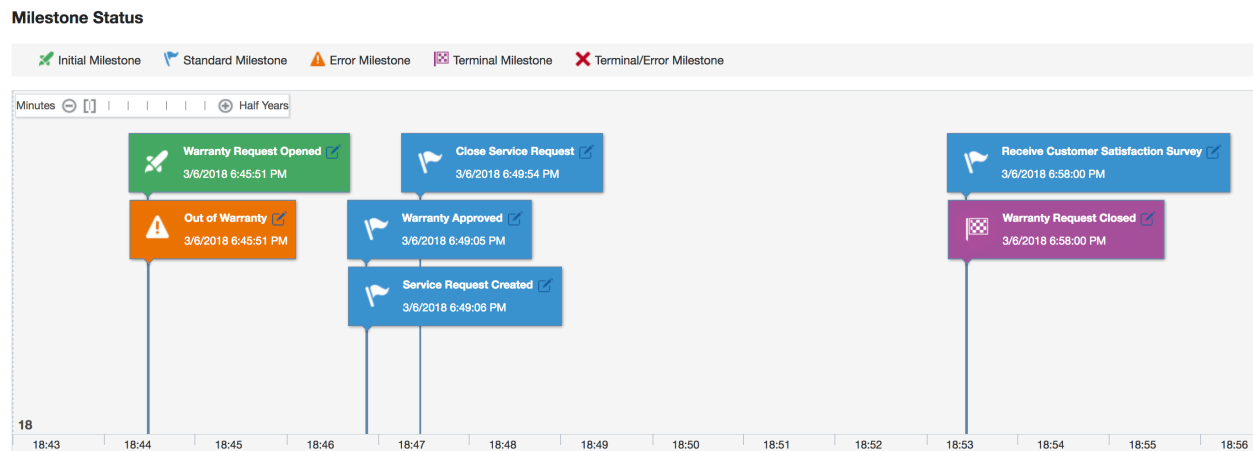
8. Double click the bar representing your name in the *Warranty Claims by Retailer* dashboard.



9. Click the first line item in the *Instances of Warranty Claim – Process Lab* page.

Warranty Request Id	Warranty Start Date	Warranty End Date	Warranty Status	Last Milestones	Warranty Number
c29377ce-89c5-573b-ec...	03/06/2018 18:45:51	03/06/2018 18:58:00	Successful	Warranty Request Closed	Nathan Angstad...

10. Review the Milestone Status of your warranty claim.



**Oracle Corporation, World Headquarters**

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle.com/oracle



facebook.com/oracle



twitter.com/oracle



oracle.com

ORACLE INTEGRATION CLOUD

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0116

Integrate, Automate and Analyze

March 2018

Author: Nathan Angstadt



Oracle is committed to developing practices and products that help protect the environment