

Exercise 9-1: Configure Post-Installation Steps and Validate Apex Classes

Scenario



Eliza Young, Infiwave's Developer, is quickly learning that with all the products Infiwave offers and the possible combinations of those products the product catalog is going to grow rapidly. She's concerned about the increased time that will be needed on maintenance and administration. She thinks to herself, there must be a more efficient way to price the hundreds of products in the catalog.

In her search for a solution, she comes across attribute-based pricing. Implementing attribute-based pricing allows Eliza to create a single product with a set of attributes, and use a matrix to price each combination of attributes. Hooray! Eliza is ecstatic that less time will be spent on maintaining and administering the product catalog - and she's also delighted to have her favorite Pricing Specialist (you) available to support her.

Goals

- Review the DefaultPricingPlan steps
- Review the PricingPlanService implementation
- Review the PricingPlanHelper Apex class
- Explain the concept of attribute-based pricing
- Implement the three key types of attribute-based pricing
- Set up attribute-based pricing using pricing plans
- Add data to lookup tables
- Create expression sets to price the line item in the cart
- Add a new step to the default pricing plan

Tasks

1. Configure post-installation steps
2. Validate Apex classes

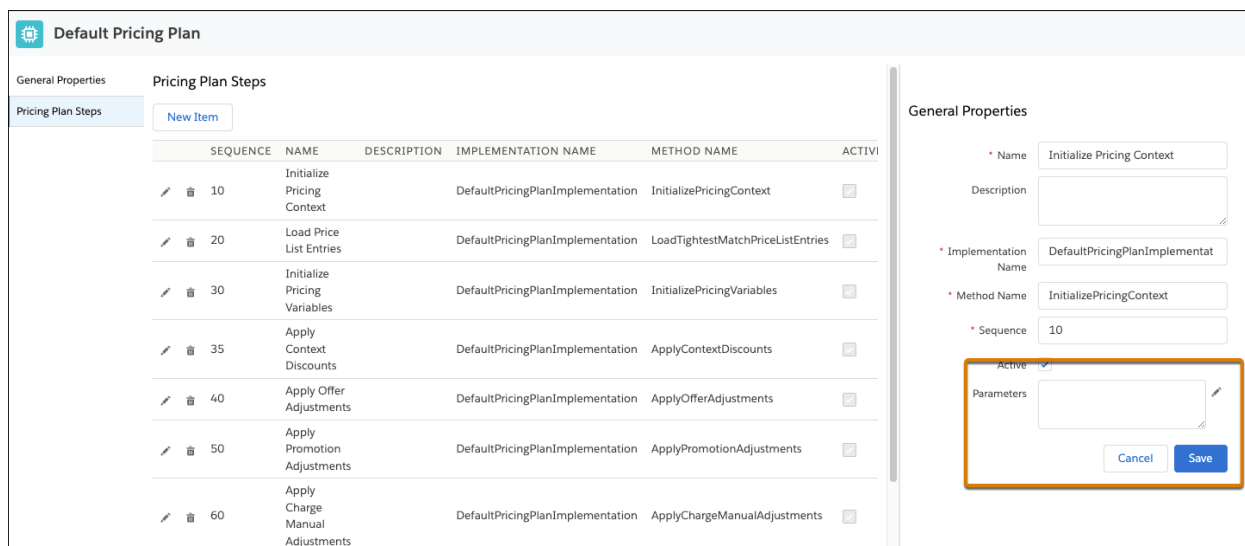
Time: 5 mins

Task 1: Configure post-installation steps

In her enthusiasm to get started, Eliza has already downloaded and installed the Attribute-Based Pricing DataPack from the [Salesforce Industries Process Library](#). She asks you to check if all the post-installation steps have been completed correctly.

Instructions

1. Verify the **Default Pricing Plan** is installed.
 - a. Using the App Launcher, locate and open the **Vlocity Product Console**.
 - b. Next to **Pricing Plan** in the Pricing area, click the search icon.
 - c. In the **Search Pricing Plan...** dialog box, press the return or enter key to display the list of pricing plans available. Notice that **Default Pricing Plan** is shown on the list.
2. Verify the **Parameters Field** displays for each step in the Default Pricing Plan.
 - a. Click on the **Default Pricing Plan** to open it for editing.
 - b. Click on the **Pricing Plan Steps** facet.
 - c. Click on any step.
 - d. In the **General Properties** pane, verify the **Parameters** field displays.



| SEQUENCE | NAME | DESCRIPTION | IMPLEMENTATION NAME | METHOD NAME | ACTIV |
|----------|---------------------------------|-------------|----------------------------------|-----------------------------------|-------------------------------------|
| 10 | Initialize Pricing Context | | DefaultPricingPlanImplementation | InitializePricingContext | <input checked="" type="checkbox"/> |
| 20 | Load Price List Entries | | DefaultPricingPlanImplementation | LoadTightestMatchPriceListEntries | <input checked="" type="checkbox"/> |
| 30 | Initialize Pricing Variables | | DefaultPricingPlanImplementation | InitializePricingVariables | <input checked="" type="checkbox"/> |
| 35 | Apply Context Discounts | | DefaultPricingPlanImplementation | ApplyContextDiscounts | <input checked="" type="checkbox"/> |
| 40 | Apply Offer Adjustments | | DefaultPricingPlanImplementation | ApplyOfferAdjustments | <input checked="" type="checkbox"/> |
| 50 | Apply Promotion Adjustments | | DefaultPricingPlanImplementation | ApplyPromotionAdjustments | <input checked="" type="checkbox"/> |
| 60 | Apply Charge Manual Adjustments | | DefaultPricingPlanImplementation | ApplyChargeManualAdjustments | <input type="checkbox"/> |

General Properties

Name: Initialize Pricing Context

Description: [Empty]

Implementation Name: DefaultPricingPlanImplementat

Method Name: InitializePricingContext

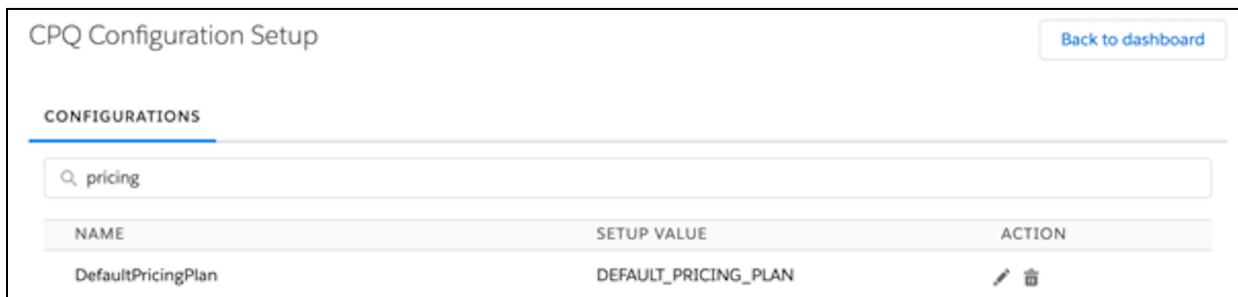
Sequence: 10

Active:

Parameters: [Empty]

Buttons: Cancel, Save


3. Verify the **DefaultPricingPlan** custom setting has been created. This custom setting is used to indicate the default pricing plan used in the org.
 - a. Using the App Launcher, locate and select **CME Admin**.
 - b. On the **Vlocity CMT Administration** tab, under **Custom Settings**, click **CPQ Configuration Setup**.
 - c. In the Search dialog box under **CONFIGURATIONS**, enter `pricing`.
 - d. The **DefaultPricingPlan** custom setting displays. Notice that the **SETUP VALUE** is set to the pricing plan's code.



CPQ Configuration Setup Back to dashboard

CONFIGURATIONS

Q pricing

| NAME | SETUP VALUE | ACTION |
|--------------------|----------------------|---|
| DefaultPricingPlan | DEFAULT_PRICING_PLAN |   |

4. Review the **PricingPlanService** implementation.



The PricingPlanService implementation is required when using pricing plans. It is the same implementation as the PricingElementServiceImplementation but the PricingPlanService allows you to manage all of its pricing steps using a pricing plan defined in Vlocity Product Console.

- a. Select the **Interface Implementations** tab.
- b. Change the List View to **All**.
- c. Use the search to locate `PricingInterface`, then select **PricingInterface** to open it for editing.
- d. On the **Related** tab, verify the **Active** and **Default** boxes are checked for the **PricingPlanService**.

Interface Implementation
PricingInterface

Related Details

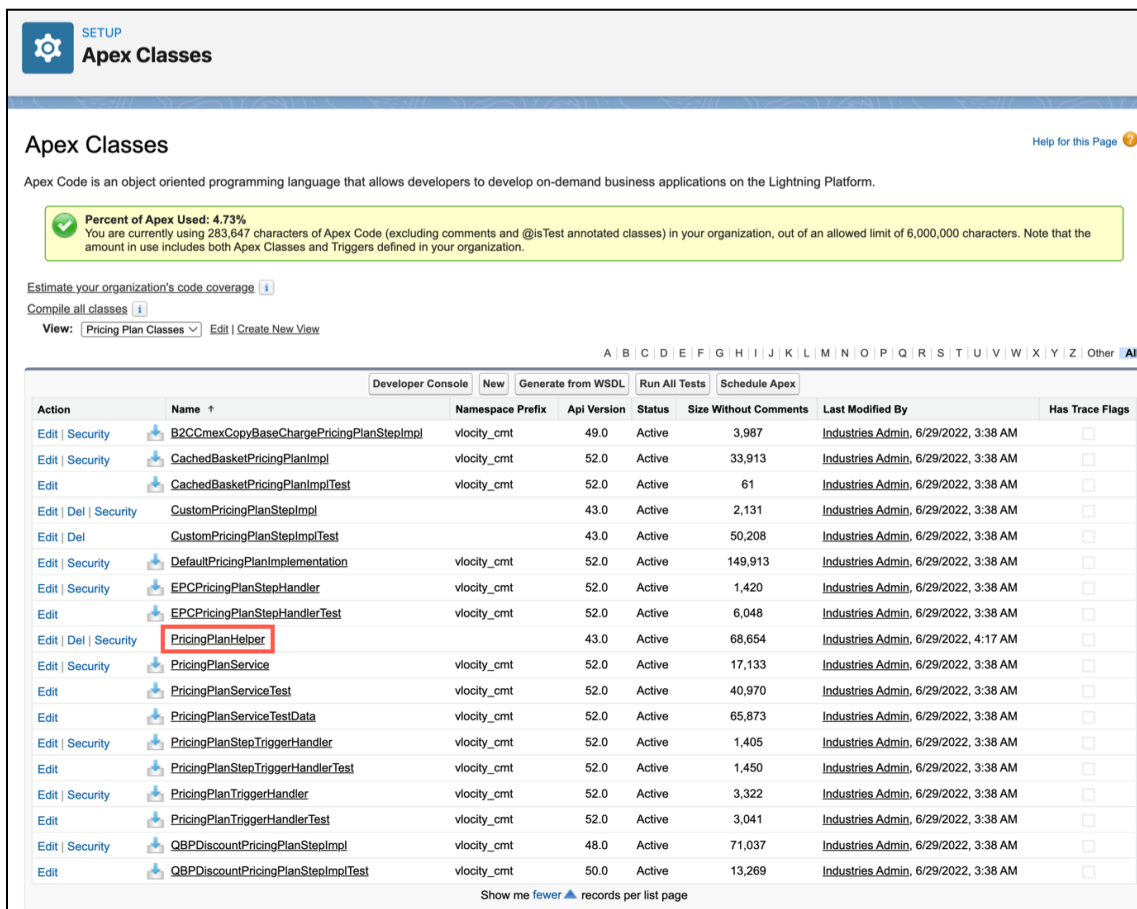
Interface Implementation Detail (4)

| Available Implementation | Active | Default |
|---|-------------------------------------|-------------------------------------|
| DefaultPricingImplementa... | <input type="checkbox"/> | <input type="checkbox"/> |
| PricingRulesImplementation | <input type="checkbox"/> | <input type="checkbox"/> |
| PricingElementServiceImp... | <input type="checkbox"/> | <input type="checkbox"/> |
| PricingPlanService | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Task 2: Validate Apex classes

Next, Eliza will verify the **PricingPlanHelper.cls** Apex class from the DataPack is installed. This Apex class is invoked by CustomPricingPlanStepImpl.cls which analyzes the matrix and invokes the calculation procedure or expression set. Due to code dependencies, when setting up these Apex classes they must be created in the order demonstrated.

1. From the Salesforce **Setup** menu, select **Setup**.
2. In the **Quick Find** search box, enter and select **Apex Classes**.
3. On the Apex Classes page, verify the **View** is set to **Pricing Plan Classes**.
4. Verify **PricingPlanHelper** displays in the **Name** column.



Apex Classes

Apex Code is an object oriented programming language that allows developers to develop on-demand business applications on the Lightning Platform.

Percent of Apex Used: 4.73%
You are currently using 283,647 characters of Apex Code (excluding comments and @isTest annotated classes) in your organization, out of an allowed limit of 6,000,000 characters. Note that the amount in use includes both Apex Classes and Triggers defined in your organization.

Estimate your organization's code coverage [i](#)
Compile all classes [i](#)

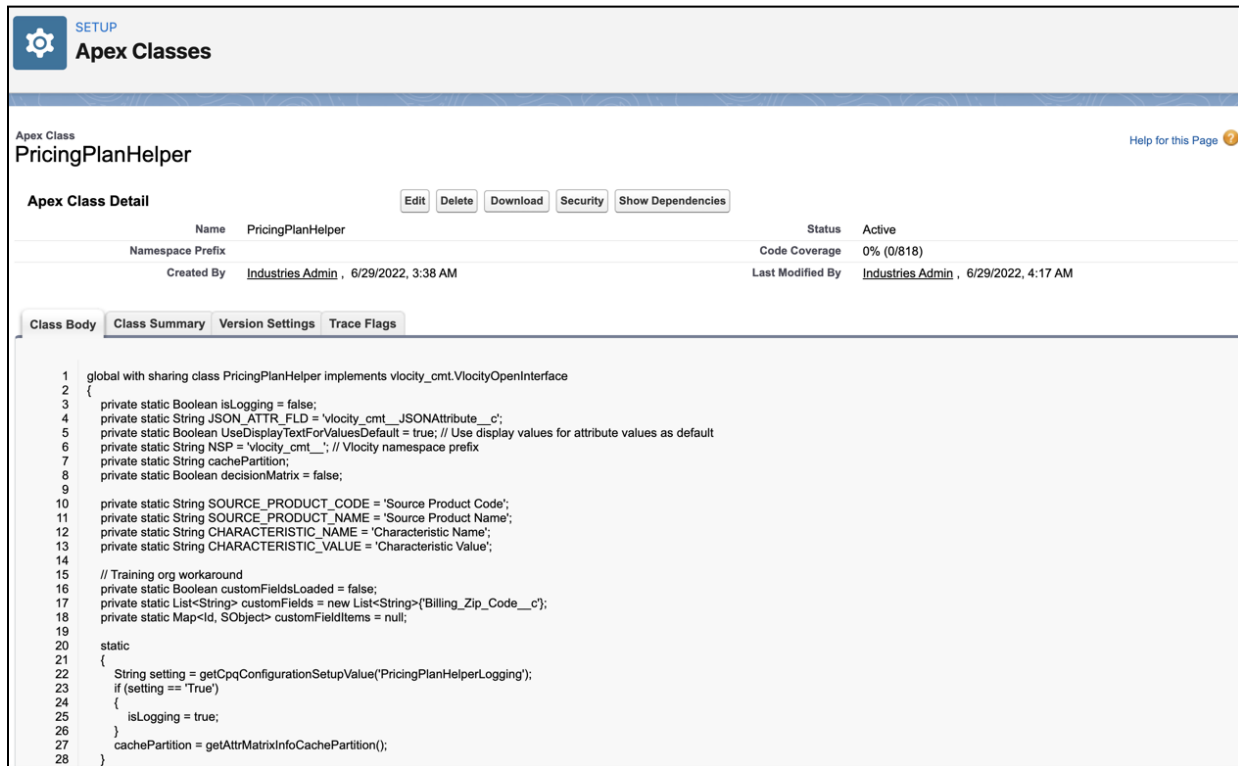
View: Pricing Plan Classes [Edit](#) [Create New View](#)

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | **All**

| Action | Name ↑ | Namespace Prefix | Api Version | Status | Size Without Comments | Last Modified By | Has Trace Flags |
|---|--|------------------|-------------|--------|-----------------------|--------------------------------------|--------------------------|
| Edit Security | B2CCmexCopyBaseChargePricingPlanStepImpl | vlocity_cmt | 49.0 | Active | 3,987 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Security | CachedBasketPricingPlanImpl | vlocity_cmt | 52.0 | Active | 33,913 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit | CachedBasketPricingPlanImplTest | vlocity_cmt | 52.0 | Active | 61 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Del Security | CustomPricingPlanStepImpl | | 43.0 | Active | 2,131 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Del | CustomPricingPlanStepImplTest | | 43.0 | Active | 50,208 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Security | DefaultPricingPlanImplementation | vlocity_cmt | 52.0 | Active | 149,913 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Security | EPCPricingPlanStepHandler | vlocity_cmt | 52.0 | Active | 1,420 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit | EPCPricingPlanStepHandlerTest | vlocity_cmt | 52.0 | Active | 6,048 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Del Security | PricingPlanHelper | | 43.0 | Active | 68,654 | Industries Admin, 6/29/2022, 4:17 AM | <input type="checkbox"/> |
| Edit Security | PricingPlanService | vlocity_cmt | 52.0 | Active | 17,133 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit | PricingPlanServiceTest | vlocity_cmt | 52.0 | Active | 40,970 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit | PricingPlanServiceTestData | vlocity_cmt | 52.0 | Active | 65,873 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Security | PricingPlanStepTriggerHandler | vlocity_cmt | 52.0 | Active | 1,405 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit | PricingPlanStepTriggerHandlerTest | vlocity_cmt | 52.0 | Active | 1,450 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Security | PricingPlanTriggerHandler | vlocity_cmt | 52.0 | Active | 3,322 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit | PricingPlanTriggerHandlerTest | vlocity_cmt | 52.0 | Active | 3,041 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit Security | QBPDDiscountPricingPlanStepImpl | vlocity_cmt | 48.0 | Active | 71,037 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |
| Edit | QBPDDiscountPricingPlanStepImplTest | vlocity_cmt | 50.0 | Active | 13,269 | Industries Admin, 6/29/2022, 3:38 AM | <input type="checkbox"/> |

Show me [fewer](#) records per list page

5. Right-click to open the **PricingPlanHelper** in a new tab to review the Apex code.



Apex Classes

Apex Class: PricingPlanHelper

Apex Class Detail

| | | | |
|------------------|--------------------------------------|------------------|--------------------------------------|
| Name | PricingPlanHelper | Status | Active |
| Namespace Prefix | | Code Coverage | 0% (0/818) |
| Created By | Industries Admin, 6/29/2022, 3:38 AM | Last Modified By | Industries Admin, 6/29/2022, 4:17 AM |

Class Body | Class Summary | Version Settings | Trace Flags

```
1 global with sharing class PricingPlanHelper implements vlocity_cmt.VlocityOpenInterface
2 {
3     private static Boolean isLogging = false;
4     private static String JSON_ATTR_FLD = 'vlocity_cmt_JSONAttribute__c';
5     private static Boolean UseDisplayTextForValuesDefault = true; // Use display values for attribute values as default
6     private static String NSP = 'vlocity_cmt__'; // Vlocity namespace prefix
7     private static String cachePartition;
8     private static Boolean decisionMatrix = false;
9
10    private static String SOURCE_PRODUCT_CODE = 'Source Product Code';
11    private static String SOURCE_PRODUCT_NAME = 'Source Product Name';
12    private static String CHARACTERISTIC_NAME = 'Characteristic Name';
13    private static String CHARACTERISTIC_VALUE = 'Characteristic Value';
14
15    // Training org workaround
16    private static Boolean customFieldsLoaded = false;
17    private static List<String> customFields = new List<String>('Billing_Zip_Code__c');
18    private static Map<Id, SObject> customFieldItems = null;
19
20    static
21    {
22        String setting = getCpqConfigurationSetupValue('PricingPlanHelperLogging');
23        if (setting == 'True')
24        {
25            isLogging = true;
26        }
27        cachePartition = getAttrMatrixInfoCachePartition();
28    }
```

a. Review lines 100-103. What is the name of the method that retrieves the price from the expression set (calculation procedure)?

b. Review lines 375-422 and lines 441-492. What is the code doing?

c. Review lines 1329-1360, notice the code is invoking an expression set (calculation procedure). We will explore this further.

d. Close the browser tab with the PricingPlanHelper Apex code.

6. Verify the **CustomPricingPlanStepImpl.cls** and **CustomPricingPlanStepImplTest.cls** Apex classes from the DataPack are installed. The **CustomPricingPlanStepImpl.cls** Apex class exposes the **GetMatrix** method and invokes the **PricingPlanHelper** class. The **CustomPricingPlanStepImplTest.cls** is an Apex test class.

- a. In the Apex Classes list, can you see the **CustomPricingPlanStepImpl** and **CustomPricingPlanStepImplTest** in the **Name** column?
- b. Right-click to open the **CustomPricingPlanStepImpl** in a new tab to review the Apex code.
- c. Review line 8 of the code. Notice the method name. We will explore this further later in this exercise.
- d. Close the browser tabs with the Apex codes.



Yay! All done!