CPQ February 2017 Administrator Guide

06 March 2017

Table of Contents

Apttus CPQ enables sales channels to **sell more**, sell **bigger**, and sell **anywhere**. Apttus CPQ is designed to boost sales effectiveness by guiding users to the best possible quotes for all products, pricing, and channels. In addition, by integrating quoting software with your contract processes, you can automate renewal quotes based on previously agreed upon pricing and terms.

Apttus CPQ is a key component of the Apttus Quote-to-Cash solution, allowing sales organizations to speed time to revenue, prevent revenue leakage, and reduce risk.

- **Enhance Sales Effectiveness with System-Driven Guidance-** Improve deal size and close rates by providing the latest product and pricing insights to all channels.
- **Quickly Scale Your Business-** Leverage your Salesforce investment to drive productivity and global growth.
- **Sell Anything, From Anywhere, On Any Device-** With mobile capability, sales channels can close deals anytime and anywhere.

Let's Get Started!

Key Benefits

- **Boost Revenue and Maximize Margins-** Create more intelligent quotes.
- **Increase Sales Effectiveness-** Shorten cycles and reduce sales friction.
- **Streamline-** Improve customer relationships.
- **Scalability-** Rapidly grow your business with the most advanced cloud platform.

Key Capabilities

- **Advanced Configuration -** Easily configure any product or service, including bundles, options, accessories and sub-assemblies, utilizing powerful rules and constraints.
- **Advanced Pricing -** Manage complex pricing hierarchies and tiers based on different parameters, and efficiently change pricing across your configuration.
- **Catalog Management -** Efficiently manage content on product life cycles, including new product introductions and end-of life products.
- **Data-Driven Recommendations -** Recommend products based on past purchases and buying trends, to drive up-sell and cross-sell opportunities.
- **Asset-Based Ordering -** Quote from current agreements, based on existing pricing and assets being used by clients.
- **Proposal Generation -** Generate professional, compelling proposals from templates and standard terms.
- **Contract Integration -** Acquire visibility into the latest contract terms, pricing and expiration dates.
- **Multichannel Support -** Deliver CPQ functionality to partners, direct sales, call centers and e-commerce sites.
- **Self-Service -** Have partners and end customers configure orders on their own
- **eSignature -** Automate signatures and eliminate the use of paper documents.
- **Mobile and Offline Support -** Quote from anytime, anywhere, from any device.

Overview

Apttus Configuration & Pricing (CPQ) gives you visibility and control of the quote-to-order phase of your sales process. Beginning in an opportunity, you can select products or services, configure and price them for simple or highly complex scenarios, and produce high quality customer facing quotes and proposals. Apttus CPQ is a web-based, on-demand application that is accessed via a standard web browser through the Internet. Before you install and start using CPQ, you must ensure the minimum system requirements listed in this section.

Rapidly Scale your Business with Apttus CPQ

Leverage the Apttus CPQ to rationalize and scale business models and selling processes.

- **Integrate processes:** Have information flow automatically across CPQ functions including configuration, pricing, quoting, contracts and order capture, without repeatedly re-entering data. Exploit workflows to enforce process integrity.
- **Rapidly orchestrate CPQ practices:** Create, launch and coordinate product, pricing and promotional programs across all channels, globally. Easily propagate all necessary rules and workflows to every relevant customer touch point.
- **Easy-to-learn user interfaces:** Capitalize on your familiar CRM interface to drive high user adoption rates. Shorten times for sales staff and partners to become productive through better user experiences.
- **Minimal IT involvement:** With point and click administration of data, catalogs workflows and functionality, enterprises can quickly model and automate processes without relying on IT resources.

About This Guide

Apttus Configuration and Pricing (CPQ) Admin Guide is designed to provide administrators with information on configuring Apttus CPQ. This guide covers the most common use cases for Apttus CPQ Administration and assumes a level of familiarity with basic Salesforce. See the Salesforce User Guide for details on standard Salesforce functionality.

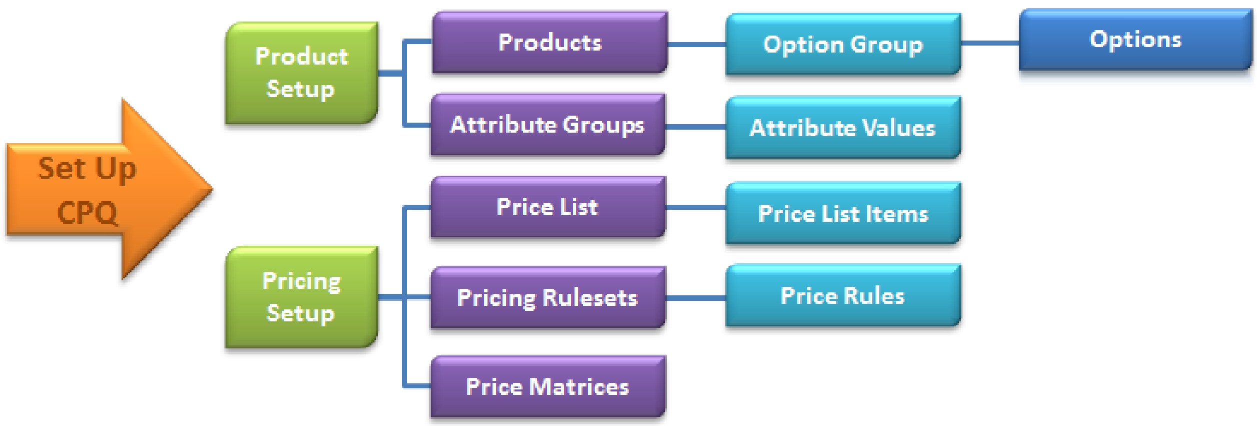
System Requirements

Before you install and start using CPQ, you must ensure the minimum system requirements.

System Requirements	
Item	**Requirement**
Operating System	Standard Salesforce.com requirements. See Salesforce PDF.
Browser	Standard Salesforce.com requirements. See Salesforce PDF.
Microsoft Office	• Microsoft Word 2003 with Service Pack 3 • Microsoft Word 2007 with Service Pack 2 plus hotfix package 976477 (Word 2007 version number has to be greater than 12.0.6520.4999) • Microsoft Word 2010 (32 bit & 64 bit)

About the Apttus CPQ Model

It is important to understand Apttus product and pricing objects and their relationships to each other in the overall application. Apttus CPQ is divided into two major areas - **Product Setup** and **Pricing Setup**. Products and Pricing are defined and customized based on how your organization wants to set up their configuration.



Logging in to CPQ

Log in to your Salesforce.com org to access Apttus CPQ.

Note

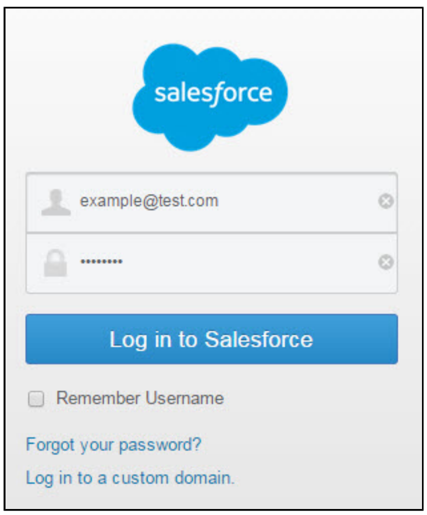
Do not use the Back button on your browser when using CPQ.

Before you log in to CPQ, make sure you meet the following criteria.

- You have installed all of the required CPQ module packages.
- You have administrative privileges.
- You have login credentials provided by Apttus.

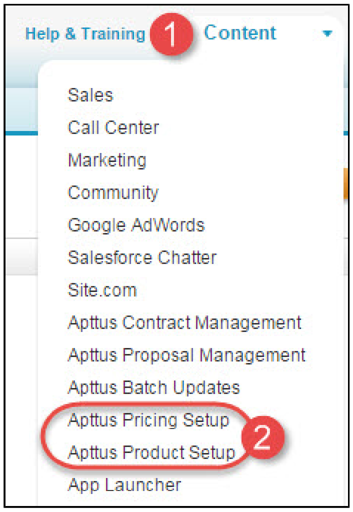
To log in to CPQ

1. Go to http://www.salesforce.com.
   Or
   If your organization is using a sandbox or test environment to access Apttus CPQ (for example, if you are doing user acceptance testing), go to http://test.salesforce.com instead.
2. In the toolbar at the top of the page, click **Login**. The login page opens.



3. Enter your User Name and Password, and click **Log in to Salesforce**.
   You have successfully logged into CPQ.

4. Navigate to the Force.com App Menu and select either **Apttus Product Setup** or **Apttus Pricing Setup** to access the Product or Pricing setup tabs, respectively.



Now you are ready to start configuring your products and pricing in Apttus CPQ.

Creating an Opportunity in Salesforce

Opportunities are the qualified contacts or accounts that you have already talked to and have entered into your sales cycle. Adding opportunities to Salesforce builds your pipeline and increases your sales forecast. You will use the Salesforce Opportunities tab to create and find your opportunities.

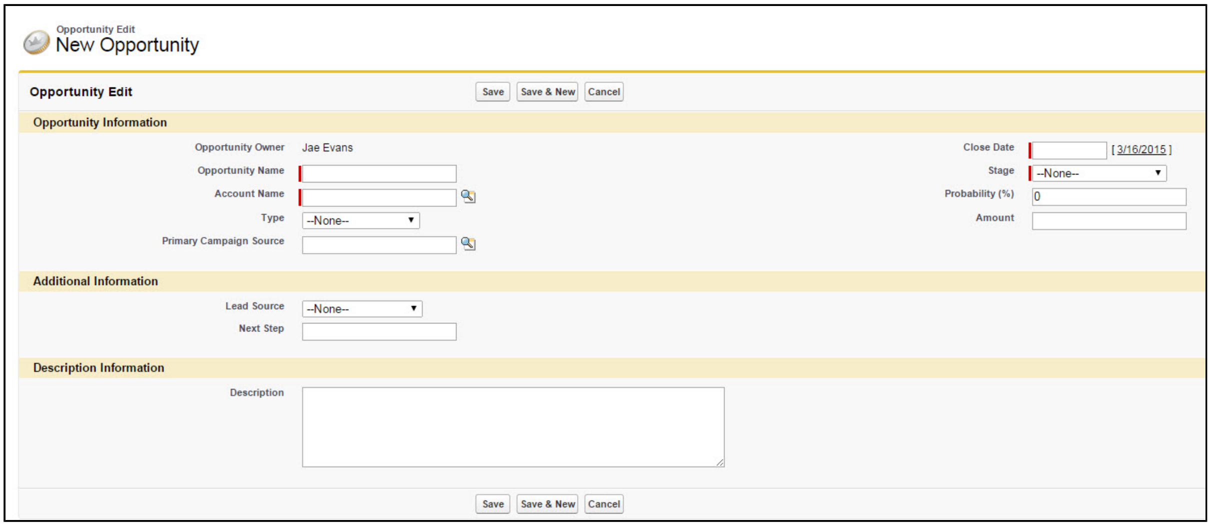
The CPQ process begins as soon as you create an opportunity.

> Opportunities are standard Salesforce objects, not Apttus custom objects. For more information on opportunities, see Salesforce documentation.

To create an opportunity

1. Log in to Salesforce.com.

2. Navigate to the **Opportunities** tab, and click **New**.



3. Enter the **Opportunity Name, Account Name, Close Date,** and **Stage**.
4. Enter additional information as required.
5. Click **Save**.

You have created an opportunity. For complete information about creating opportunities in Salesforce, refer to Create Opportunities and Opportunity Fields in the Salesforce documentation.

Upgrade Information

Before You Upgrade

Before you upgrade, it is important that you know about all of the changes that have occurred between your existing release and the release you are upgrading to. To be sure you have all the information you need, refer the release notes for all intermediate releases.

Previous Production releases:

- CPQ November 2016 Release Notes
- CPQ August 2016 Release Notes
- CPQ May 2016 Release Notes
- CPQ February 2016 Release Notes
- CPQ November 2015 Release Notes
- CPQ August 2015 Release Notes
- CPQ May 2015 Release Notes
- CPQ February 2015 Release Notes
- CPQ SP8 Release Notes
- CPQ 7.0 Release Notes

See the release notes for packages required to upgrade. For complete installation information, see Installing Apttus CPQ.

After You Upgrade

Object Maintenance

After upgrading your packages, you must run all the object maintenance tasks to synchronize all your previous configurations.

To access the maintenance pages, click ✚ (All Tabs) to display all tabs and click the link for each required maintenance. When you run maintenance jobs after an upgrade, you must run them in the order below, in the instances described for each:

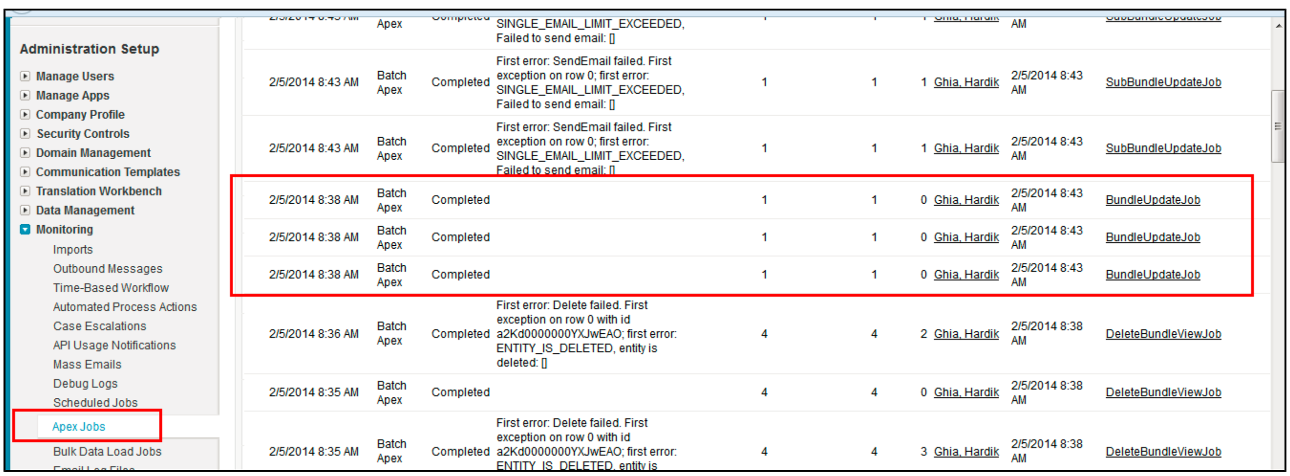
Product Filter Maintenance - After every upgrade, you must run this task, which collects all product filter field values related to a category. The maintenance job updates the required records for all the categories every time you run it. You must also run the maintenance task every time there is a change in products or a change in the filter field values.

Category Maintenance - After any product association, removal or hierarchy change, you must run a Category Maintenance job. This de-normalizes the hierarchy into a custom object for reporting and totaling purposes. If category maintenance is not run, the end user may see incorrect totals on the shopping cart page.

Constraint Rule Maintenance - This should be run after any changes are made to Constraint Rule conditions. If it isn't run, products may not be included or excluded as expected.

Bundle Maintenance - When options are added or removed from a bundle, you must use the Update Bundle Components page to run a bundle maintenance job that synchronizes all the bundles and options.

When you run bundle maintenance for all bundles, be sure to only click the button once to avoid scheduling multiple unnecessary bundle maintenance jobs. To track the progress of the job, you should go to **Setup > Administration Setup > Monitoring > Apex Jobs**. You see the bundle jobs and whether they have been completed.



Attribute Maintenance - After an attribute group is associated with a category, run this job to associate the attribute with the products that belong to the category.

Criteria Maintenance - After any pricing change (rules, dimensions or any criteria change), you must run a Criteria Maintenance job.

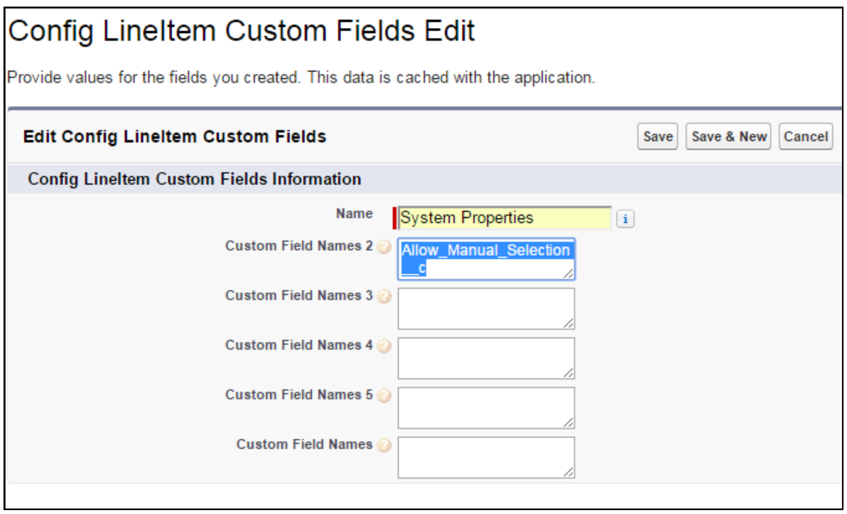
Ensure that you run the Criteria Maintenance job only if you have active rulesets or have made any updates to the active rulesets.

Adding custom fields to Config LineItem Custom Fields

If your implementation uses the custom callback class related to Pricing, Related Pricing, Asset Line Item, and Validation, then the API names of the custom fields referred in the custom callback class must be defined in the following custom setting:

Config LineItem Custom Fields

If the length of the text field exceeds the first column, use the next text fields for this.



Installing Apttus CPQ

This topic contains the following subtopics:

- Before You Install Apttus CPQ—Before you install Apttus CPQ module packages, you must ensure that Salesforce CRM Content is enabled in your org. This section contains the procedure for verifying your Salesforce CRM Content is enabled.
- Installing the Apttus CPQ Packages—Multiple packages must be installed to implement the complete CPQ solution. This section provides installation instructions, including the required installation sequence of Apttus packages.
- Recommended Post-Installation Configuration—Although your CPQ implementation will work well with the out-of-the-box configuration, you can use custom settings to configure the system to meet your specific business requirements. This section contains procedures for configuring a variety of system properties and settings.

Before You Install Apttus CPQ

Before you can install Apttus CPQ module packages, you must ensure that Salesforce CRM Content is enabled in your org.

To enable Salesforce CRM Content

You must be logged into Salesforce.com.

1. Go to **Setup > Customize > Salesforce Files > Settings > Salesforce CRM Content** and click **Edit**.
2. Select the **Enable Salesforce CRM Content** check box.
3. Click **Save**. Salesforce CRM Content is enabled.

Now, you can start installing the Apttus CPQ packages.

Installing the Apttus CPQ Packages

Multiple packages must be installed to implement the complete CPQ solution. Packages for CPQ must be installed in the order indicated in the table in this section. You begin with the Apttus base packages and then install the integration packages that enable the various products to function together.

You may not need to install all of the packages. For example, if your implementation does not use Apttus Advanced Approvals you do not need to install Apttus Approvals Management or Apttus Quote/Proposal Approvals Management. Refer to the table in this section to learn which packages are required for standalone CPQ and which packages are required only when you are using other Apttus products.

⚠ Caution

Apttus recommends downloading and upgrading Apttus packages in a Salesforce sandbox **before** installing them in your production environment. For information on installing and upgrading in a sandbox, please contact Apttus Support before you install any packages.

Install the packages in the following order.

Order	Package	Install Center tab to access the package	Required?	Install this package only if you also have...
1	Apttus Contract Management	Contract Management	Y	
2	Apttus Proposal Management	CPQ	Y	
3	Apttus Configuration & Pricing	CPQ	Y	
4	Apttus Quote/Proposal-Configuration Integration	Integrations	Y	

5	Apttus CPQ API	CPQ	Y	
6	Apttus Quote/Proposal-Contract Integration	Integrations	N	Contract Management
7	Apttus Contract-Configuration Integration	Integrations	N	Contract Management
8	Apttus Quote/Proposal-Asset Integration	Integrations	N	Renewals

You can additionally install the following packages if you are using the new Deal Maximizer Setup and/or CPQ Header Level Approvals.

- Apttus Deal Maximizer Setup
- Apttus Deal Maximizer
- Apttus CPQ Maximizer

If you are using Advanced Approvals for Quote, you must install the following packages:

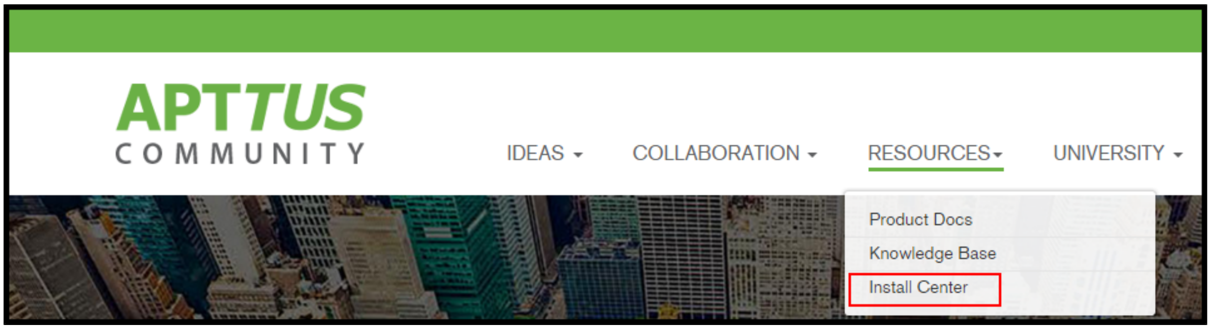
- Advanced Approvals Management
- Apttus CPQ Approvals Management
- Apttus Quote/Proposal Approvals Management (This package is not required if you are not using Approvals for Agreements)

Note

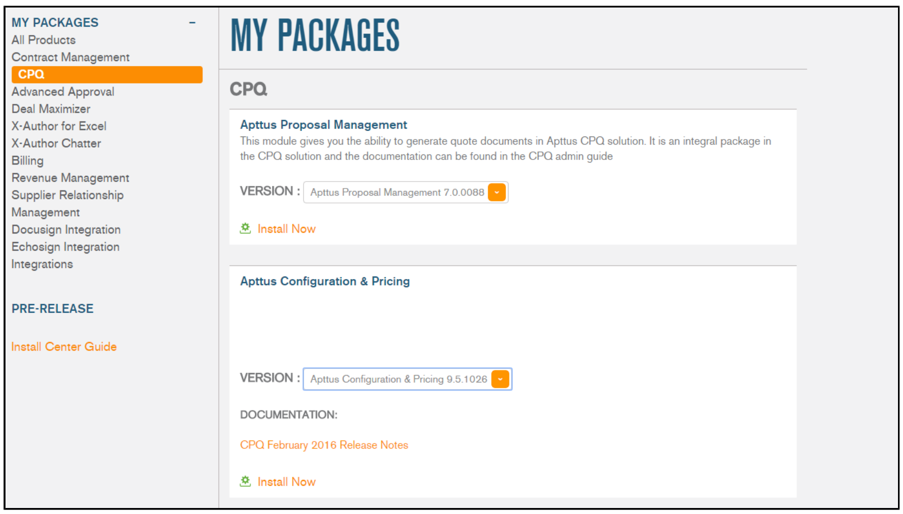
You must have Apttus-provided login credentials to the Apttus Community Portal to be able to download packages.

To install the CPQ module packages

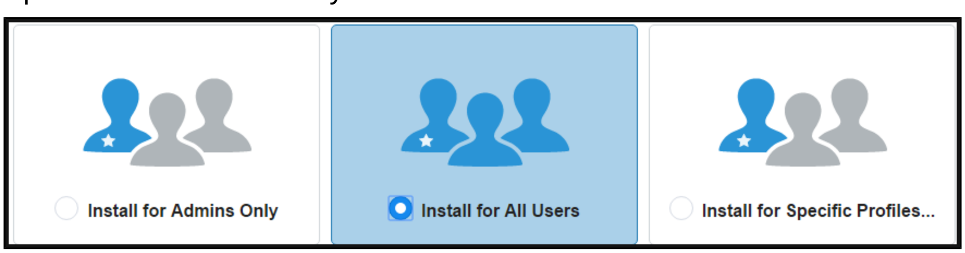
1. Go to the **Install Center** tab on the Apttus Community Portal.



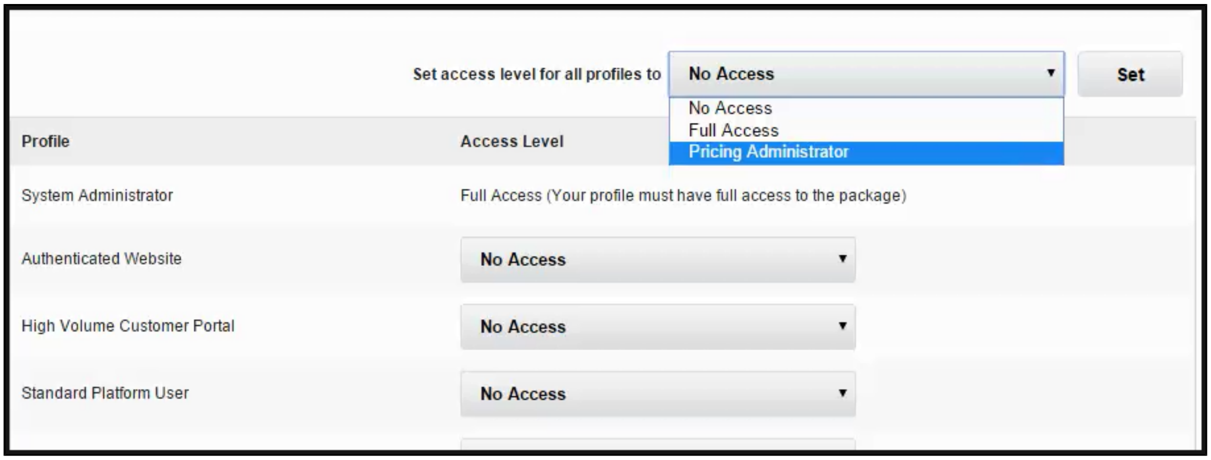
2. In **My Packages** navigation link, click **CPQ**. From the **VERSION** drop-down, select the version that you want to install.



3. Click **Install Now.**
4. Select the environment in which you want to install the packages. Click **Install in Production** to install the packages in your production environment. Click **Install in Sandbox** to install the packages in your sandbox.
5. In the Salesforce login screen, enter your login credentials and click **Log In**.
6. On the Upgrade page, enter the password provided by Apttus.
7. Apttus recommends that you select **Install for All Users**.



8. If you want to **Install for Specific Profiles**, you must define the access level for all profiles. Select from one of the following options.



- **No Access** - This is the default setting. Apply this access level to disable all object permissions.
- **Full Access** - Apply this access level to assign users permissions to Read, Create, Edit, Delete,View All, and Modify All for all objects in the CPQ package.
- **Pricing Administrator** - Apply this access level to assign users permissions to Read, Create, Edit, Delete,View All, and Modify All for all pricing-related objects in the CPQ package.

> ### Note
> If a permission is not enabled for a profile, you can apply Apttus-provided permission sets such as the **Apttus CPQ Admin** Permission Set and **Apttus CPQ User** Permission Set to enhance the access levels of the user profile. To assign a permission set , select it from the Available **Permission Sets** box and click **Add**. To remove a **permission set assignment** , select it from the **Enabled Permission Sets** box and click **Remove**. For more information, see the Salesforce Administrator Guide.

9. Click **Set**.
10. Click **Upgrade**

A message is displayed indicating the installation is underway. Once installed, repeat this procedure for each of the packages.

Recommended Post-Installation Configuration

Post-installation settings are part of your specific business requirements which means your CPQ implementation will work with out-of-the-box installation, but you can customize the application per your specific implementation using custom settings.

- To customize the CPQ application, use the System Properties
- To customize the product selection page, use the Catalog Page Settings
- To customize the option selection pages, use the Bundle Page Settings
- To customize the Installed Products page, use the Installed Products Settings
- To customize Apttus Quote/Proposal, use the Proposal System Properties
- To customize Comply, use the Comply System Properties
- To support product and proposal summary in Salesforce Lightning, use the Object Summary Settings

Customizing the CPQ Application

To customize the Apttus CPQ application for your business requirements, use the **System Properties** config settings.

To customize the CPQ application

1. Click the All Tabs icon ✚ and click **Config Settings**.
2. Click **System Properties**.
3. Enter values as required in one or more of the fields and click **Save**.

Customizing the Catalog Page

To customize the Catalog page for product selection for your business requirements, use the **Catalog Page Settings**.

To customize the catalog page

1. Click the All Tabs icon ✚ and click **Config Settings**.

2. Click **Catalog Page Settings**.
3. Enter values as required in one or more of the fields and click **Save**.

The new settings are applied and the catalog page is displayed.

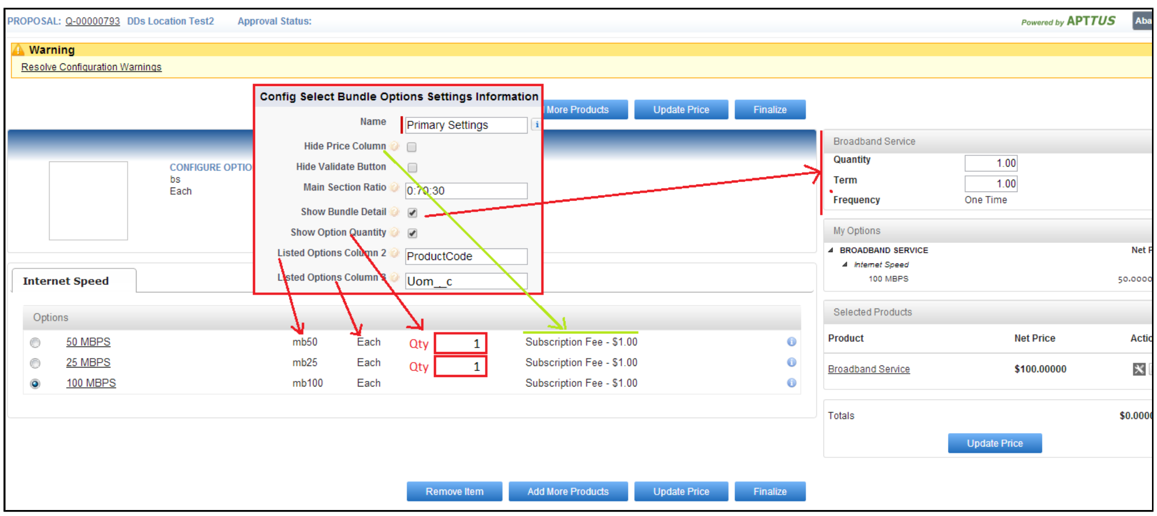
Customizing the Options Page

To customize the options page for your business requirements, use the **Bundle Page Settings**.

To customize the options page

1. Click the All Tabs icon ✚ and click **Config Settings**.
2. Click **Bundle Page Settings**.
3. Enter values as required in one or more of the fields and click **Save**.

The new settings are applied and the select bundle options page is displayed.



Customizing the Installed Products Page

To customize the Installed Products page for your business requirements, use the **Installed Products Settings**.

To customize the installed products page

1. Click the All Tabs icon ✚ and click **Config Settings**.

2. Click **Installed Products Settings**.
3. Enter values as required in one or more of the fields and click **Save**.

Customizing the Quote/Proposal Application

To customize the Apttus Quote/Proposal application for your business requirements, use the **Proposal System Properties** custom setting.

To customize the quote/proposal application

1. Go to **Setup > App Setup > Develop > Custom Settings**.

1. Click **Manage** for **Proposal System Properties** and click **New**.
2. In **Name**, type ***System Properties*** as the data set name.
3. Enter details in one or more of the required fields and click **Save**.

Customizing the Comply Application

To customize the Apttus Comply (Contract Management) application for your business requirements, use the **Comply System Properties** custom setting.

To customize the comply application

1. Go to **Setup > App Setup > Develop > Custom Settings**.
2. Click **Manage** for **Comply System Properties** and click **New**.
3. In Name, type ***System Properties*** as the data set name.
4. Enter details in one or more of the required fields and click **Save**.

Custom Settings

This section contains information on configuring the following custom settings and system properties.

System Properties

This page lists all the custom settings you may require to set up your CPQ application. You can enter details in one or more of the following sections, based on your organization's requirement.

- Page Selection Settings
- Pricing Settings
- Constraint Rule Settings
- Cart Related Settings
- Precision Settings
- SystemProperties
- Miscellaneous Settings

Page Selection Settings

Fields	**Description**
View Cart Page	Select the default cart page to be displayed. For example: *CartDetailView*
Default Catalog Page	Select the default catalog page for product selection. For example: *SelectConfigProductsFilterView*
Options Page	Select the default options page.Tip: The *SelectConfigOptionsDetailView* page does not support more than 300 options on a bundle with Usage Tiers, Price Ramps, and Attributes for less than 300 Options on a Bundle, instead select *ConfigureBundle*.
Guide Page Default	Type the Visualforce page name for the default guide page for product selection.
Product Attribute Detail Page	Type the custom Visualforce page name to capture product attribute detail information.

Fields	Description
	For example: *Apttus_Config2__ProductAttributeDetail3*
Resolve Configuration Page	Select the default Resolve Configuration page for errors or warnings.

Pricing Settings

Setting	Description
Defer Pricing	Performs pricing of products only after you click the Go to Pricing button; whereas, clearing this check box enables the system to perform pricing as and when you add or delete products.
Enable Auto Reprice	Automatically reprices the cart. In the cart, when you modify pricing and tab out, the system automatically recalculates and displays the price.
Enable Matrix Pricing For Options	Indicates whether matrix pricing is enabled for options. This may be used to optimize performance when options do not use price matrices in the organization. The default value is false (disabled).
Enable Keyed Matrix Pricing	Select this if you have option products with large number of dimensions of type discrete. This also helps in reducing SOQL queries.
Max Adjustment Lines	Type the maximum number of discount lines allowed for a line item. The default value is one. You can create more than one adjustment only when the value is greater than one.
Related Price Scope	

Setting	Description
	Enables you to choose if you want to perform the related pricing calculations over the entire cart or confine the calculations to a Bundle product only. For example, if the price of an Option product is a percentage of another Option product in the same bundle, consider the following scenarios with **Related Price Scope** = *Cart* and *Bundle*, • If value = *Cart* , the Option price is based on the price of all the instances of the related product in the cart, whether it is in the same Bundle or a Standalone or in an another Bundle. • If value = *Bundle* , the Option price is based on the related product instances within the same Bundle.
Enable Aggregate Pricing	Indicates whether aggregate pricing is enabled. This may be used to optimize performance when aggregate price rules are not used in the organization. The default value is false (disabled).
Enable External Pricing	Indicates whether pricing will be done externally. The default value is false (disabled).
Misc Charge Types	Enter the list of miscellaneous charge types. Each charge type should be separated by a new line. The charge types are displayed as pick list values when adding miscellaneous items to the cart. The default values are *Sales Tax* and *Shipping & Handling*.
Totaling Group Type	Type the default value for summary group type picklist. Valid values are: • *Category*: When you add a product to the cart from the lowest leaf category where it is associated, the Category Hierarchy column in the **Line Items** related list is populated with its bread-crumb trail. This is useful to identify the exact category from where the product was added, in case if the product is associated with multiple categories. • *Product* The default value is *Category*. For Product-based totaling, the Product Totaling Hierarchy setting is required.

Setting	Description
Adhoc /Product Totaling Hierarchy	Enter the list of product fields that represent the product totaling hierarchy. Each product field API name should be separated by a new line. Only applicable when Product based totaling is in effect.
Compute Totals In Separate Step	Indicates whether the totaling should be performed in a separate step or it should be combined with the base pricing step. This setting ensures that the totaling, which is dependent on the number of lines, should be done as a separate remoting call. This setting helps to avoid the CPU time limit issue to a large extent.

Constraint Rule Settings

Setting	Description
Show Info In Header	Indicates whether to show information message from auto include or auto exclude in the header. The default value is false (Hide).
Hide Resolve Config Link	Hides the link to the resolve configuration error page and instead displays the error along with the constraint rule message.
Disable Constraint Rules	Indicates whether constraint rules are disabled. This may be used to optimize performance when constraints are not used in the organization. The default value is false (enabled).
Cleanup Invalid Rule Prompt	Enable removal of line items prompted by currently invalid rules.
Max Constraint Rules Round Trip	Type the maximum number of round trips after which the constraint rule processing should stop. Round trip happens only when auto included products trigger more rules. The default value is 3.
	This setting is deprecated and not functional.

Setting	Description
Defer Constraint on Option Selection	
Enable Defaulting For Products	Indicates whether defaulting is enabled for products. This may be used to optimize performance when product default rules are not used in the organization. The default value is false (disabled).
Skip Constraint Decision Field	Field on the line item when checked, skips Constraint check. Enables bypassing constraint and finalize quote even if there are constraint error.

Cart Related Settings

Setting	Description
Enable Optional Items	Indicates if your bundle/standalone/option products can be displayed as non-mandatory (optional) line items on your configuration and cart pages.
Display Cart Actions As Dropdown	Groups the line item action buttons under the Actions menu icon.
Bypass Shopping Cart	Displays the **Update Price** and **Finalize** buttons and hides the **Go To Pricing** button on all the new pages. On the configuration page, only the **Go To Pricing** button is disabled. You can still finalize the quote using the Mini cart.
Keep Abandoned Carts	Indicates whether abandoned carts should be kept. The default value is false (abandoned carts are moved to the recycle bin).

Setting	Description
Save On All Actions	Saves any changes that may have been done before navigating to any other page.
Expand Bundles in Cart	Expands all attributes and options.
Product Display Max Length	Indicates the maximum length of the Product name that can be displayed on the Cart. If the character length in your Product name exceeds the character length specified in this property, the system displays an ellipsis. Else, the system wraps the characters and adjusts the space accordingly on the Cart.
Actions Column Position	Type *Left* or *Right* to position the actions column in the cart page. By default, the Actions column is displayed on the left, but may be overridden using this setting.
Hide Subtotals In Cart	Indicates whether subtotals should be hidden in the cart view. The default value is false (show subtotals in the cart).
Hide Copy Action	Hides the action in the cart. By default, this is not selected.
Show Attributes in Cart	Shows the specified attributes applicable to the line items in the cart. Attributes appear in the product dialog.
Custom Deal Guidance Page	Type the custom page name for deal guidance.

Precision Settings

Setting	Description
Currency Field Precision	Type a number to specify decimal places for currency precision. The default currency precision is 2 decimal places.
Quantity Field Precision	Type a number to specify decimal places for quantity precision. The default quantity precision is 2 decimal places.
Percentage Field Precision	Type a number to specify decimal places for percentage precision. The default percentage precision is 2 decimal places.
Term Field Precision	Type the precision value that you want to display for the **Selling Term** column on the shopping cart. For example, if your term calculation results to 4.553412 and you have set **Term Field Precision** to *3*, the **Selling Term** column displays 4.553. The default term precision is 5 decimals. This setting is applicable only for the New UI.

System Properties

Setting	Description
Base Product Relation Field	Type the field names that associate a child product with the base product.
Run Post Finalize Trigger In Async Mode	Indicates whether the finalization task should be run asynchronously.
Fixed Button Bar	Displays a fixed bar for actions buttons on the catalog, attributes, options, cart, and installed products page.
Show Admin Sidebar	Hides the sidebar for the below mentioned Admin pages:

Setting	Description
	• CPQConsole.page • CategoryManager.page • ClassificationHierarchy.page • ConstraintRuleActionCriteriaEdit.page • ConstraintRuleConditionCriteriaEdit.page • CriteriaUpdate.page • DisplayActionSettings.page • DisplayColumnSettings.page • FeatureSet.page • HierarchyViewUpdate2.page • IconUploader.page • ManageGuidedSearchRule.page • MultipleConstraintRulesAdmin.page • PriceListItem.page • PriceRule.page • ProductConsole.page • ProductConstraintView.page • ProductDefaultRuleCriteriaEdit.page • ProductFilterMaintenance.page • RelateAttributeToProduct.page • RelateFeatureToProduct.page • RelateProductToBundle.page • RelateProductToCategory.page • RelateProductToFootnote.page • SystemProperties.page
Instance URL	

Setting	Description
	The Salesforce instance URL (for example, https://na7.salesforce.com). The instance URL is required to navigate to custom pages in the managed package.
Show Header	Displays the header tabs and the sidebar.
CSS Override	Type the name of the static resource to override CSS in the catalog, attributes, options, and cart pages.
Use Enhanced CSS	Enables you to include the enhanced CSS file on CPQ pages.

Miscellaneous Settings

Setting	Description
Option Line Item Columns	Type the field names from the Line Item object displayed in the Option Hierarchy page. To add more than one field name separate the names by a new line or a comma.
Option Product Columns	Type the field names from the Product object displayed in the Option Hierarchy page. To add more than one field name separate the names by a new line or a comma.
Cascade Shared Attribute Updates	Cascades the shared bundle attribute updates to options in the bundle while configuring a product.
Auto Finalize On Cart Approval	Automatically finalizes the cart when the quote gets approved.

Setting	Description
Search Category Default	Indicates the category to default to when searching for products using the search text box on the product selection page. The default value is *All Categories*. Select *All Products* to search all products regardless of categories.
Revalidation Product Columns	Type the API names of product object fields that you want to display on the revalidation pop-up on the cart page. You can separate each field API name either by a comma or by a new line.
Auto Sync On Cart Approval	Automatically synchronizes the cart when approved and finalized.

Config System Properties

This is the standard Salesforce Custom Settings page that lists all the legacy custom settings you may require to fulfill your specific business requirements. You can access these settings from **Setup > Develop > Custom Settings > Config System Properties**.

Important

Apttus recommends using the System Properties page from the Config Settings tab. Some of the custom settings listed on this page may require you to contact Apttus Support or Apttus Professional Services to guide you through using these settings for specific business use cases.

In your org, you may see custom settings that are not listed in this document—these custom settings are not functional and are marked as Deprecated (D).

Setting	Description
Base Product Relation Field	Type the API name of the product field that associates a child product with the base product.
Defer Pricing	

Setting	Description
	Indicates whether products are priced before displaying the cart or after displaying the cart. The default is true (defer until cart is displayed).
Disable Constraint Rules	Indicates whether constraint rules are disabled. This may be used to optimize performance when constraints are not used in the organization. The default value is false (enabled).
Enable Aggregate Pricing	Indicates whether aggregate pricing is enabled. This may be used to optimize performance when aggregate price rules are not used in the organization. The default value is false (disabled).
Enable Defaulting For Products	Indicates whether defaulting is enabled for products. This may be used to optimize performance when product default rules are not used in the organization. The default value is false (disabled).
Enable External Pricing	Indicates whether pricing will be done externally. The default value is false (disabled).
Enable Matrix Pricing For Options	Indicates whether matrix pricing is enabled for options. This may be used to optimize performance when options do not use price matrices in the organization. The default value is false (disabled).
Guide Page Default	Type the Visualforce page name for the default guide page for product selection.
Hide Subtotals In Cart	Indicates whether subtotals should be hidden in the cart view. The default value is false (show subtotals in the cart). When checked, the system hides subtotal lines in the Summary section, and still shows Category Total, One Time Total, and Grand Total.
Hide Evergreen End Date	Indicates whether the end date are hidden for Evergreen Proposals.

Setting	Description
Hide Dates For One Time Charge	Indicates whether the dates are hidden for price list items of type One-Time Charge
Instance Url	The Salesforce instance URL (for example, https://na7.salesforce.com). The instance URL is required to navigate to custom pages in the managed package.
Keep Abandoned Carts	Indicates whether abandoned carts should be kept. The default value is false (abandoned carts are moved to the recycle bin).
Max Adjustment Lines	Type the maximum number of discount lines allowed for a line item.
Max Constraint Rules Round Trip	This is the maximum number of round trips after which the constraint rule processing should stop. Round trip happens only when auto included products trigger more rules. The default value is 3.
Misc Charge Types	Enter the list of miscellaneous charge types. Each charge type should be separated by a new line. The charge types are displayed as pick list values when adding miscellaneous items to the cart. The default values are *Sales Tax* and *Shipping & Handling*.
Options Page	Type the Visualforce page name for default options page. If no value is specified, the options page defaults to the icon view. Valid values are: • SelectConfigOptions • SelectConfigOptionsTabView • SelectConfigOptionsListView
Product Attribute Detail Page	Type the Visualforce page name for default attributes page. This is the custom page to capture product attribute details.
Adhoc/Product Totaling Hierarchy	

Setting	Description
	Enter the list of product fields that represent the product totaling hierarchy. Each product field API name should be separated by a new line. Only applicable when Product based totaling is in effect. Check the custom setting **Totaling Group Type**.
Skip Constraint Decision Field	Field on the line item when checked, skips Constraint check. Enables bypassing constraint and finalize quote even if there are constraint error.
Related Price Scope	Enables you to choose if you want to perform the related pricing calculations over the entire cart or confine the calculations to a Bundle product only. For example, if the price of an Option product is a percentage of another Option product in the same bundle, consider the following scenarios with **Related Price Scope** = *Cart* and *Bundle*, • If value = *Cart* , the Option price is based on the price of all the instances of the related product in the cart, whether it is in the same Bundle or a Standalone or in an another Bundle. • If value = *Bundle* , the Option price is based on the related product instances within the same Bundle.
Search Category Default	The default search category. Indicates the category to default to when searching for products using the search text box in the product selection page. The valid values are the following: *All Categories All Products* The default value is *All Categories*. Choose *All Products* to search all products regardless of categories.
Show Info In Header	Indicates whether to show information message from auto include or auto exclude in the header. The default value is false (Hide).
	The totaling group preference. The valid values are the following:

Setting	Description
Totaling Group Type	• *Category*: When you add a product to the cart from the lowest leaf category where it is associated, the Category Hierarchy column in the **Line Items** related list is populated with its bread-crumb trail. This is useful to identify the exact category from where the product was added, in case if the product is associated with multiple categories • *Product* The default value is *Category*. For Product-based totaling, the **Product Totaling Hierarchy** setting is required.
View Cart Custom Fields 2	Enter the list of additional custom fields from the line item object displayed in the custom view cart page. Each field API name should be separated by a new line or a comma.
View Cart Custom Fields	Enter the list of custom fields from the line item object displayed in the custom view cart page. Each field API name should be separated by a new line or a comma.
View Cart Page	Type the Visualforce page name for default cart page.
View Cart Total Custom Fields	Enter the list of custom fields from the summary group object displayed in the custom view cart page. Each field API name should be separated by a new line or a comma.
Actions Column Position	Type the position of the actions column in the cart page. Valid values are: • Left • Right
Admin User	The admin user is the default owner of activities created by a user who is not allowed to be the owner (for e.g. customer portal user). Enter the admin user name as first name, last name.

Setting	Description
Auto Finalize On Cart Approval	Automatically finalizes the cart when approved and ready for finalization.
Auto Sync On Cart Approval	Automatically synchronizes the cart when approved and ready for finalization.
Bypass Sharing	Indicates whether apex code can bypass record sharing.
Bypass Shopping Cart	Displays the **Update Price** and **Finalize** buttons and hides the **Go To Pricing** button on all the new pages. On the configuration page, only the **Go To Pricing** button is disabled. You can still finalize the quote using the Mini cart.
CSS Override	Type the name of the static resource to override CSS in the catalog, attributes, options, and cart pages.
Cascade Shared Attribute Updates	Cascades the shared bundle attribute updates to options in the bundle. If this setting is un-checked, the bundle attributes will initially be cascaded to options in the bundle, but subsequent changes in the bundle attribute will not be cascaded to option attributes.
Currency Field Precision	Type a number to specify decimal places for currency precision. The default currency precision is 2 decimal places.
Custom Asset Action Label Name	Type a label name for a custom asset action.
Custom Asset Action Page	Type the Visualforce name for the custom asset action page.
Default Catalog Page	Type the Visualforce page name for the default catalog page. Valid values are: • SelectConfigProducts

Setting	Description
	• SelectConfigProductsFilterView
Defer Validation Check in Bundles	Defers the validation check in the bundle configuration step. This is selected by default.
Direct Configure Asset Actions	Opens the options page for the first bundle product that is configurable. The rest of the bundle products are added to the cart and displayed under the Selected Products section on the right hand side panel. If the bundle product is set as *Must Configure*, an exclamation icon is displayed for incomplete configuration.
Disable Charge Type Totaling	Disables the totaling of charge types.
Disable Existing Asset Pricing	Disables the existing asset pricing. If disabled, prices of existing assets are not rolled up.
Display Cart Actions As Dropdown	Displays the line item action buttons as a dropdown menu in the cart page.
Enable Auto Reprice	Automatically reprices the cart when a price component is modified.
Enable Auto Sequencing For Options	Enables auto sequencing for options. If you do not select this option, the Option Item Sequence in the shopping cart and the Line Items is based on the user selection sequence rather than the Bundle Options sequence.
Enable Location	Enables you to select location for a cart. The line items will have the current cart location assigned to them.
Fixed Button Bar	Displays a fixed bar for actions button on the catalog, attributes, options, cart, and installed products page.
Hide Asset Actions	

Setting	Description
	Type the actions to hide on the Installed Products page. Use Comma to separate values. For example: Increment, Amend, Renew, Cancel
Hide Copy Action	Hides the copy action in the cart. The default value is false (copy icon is displayed).
Hide Grand Total	Hides the grand total line on the cart page.
Hide Resolve Config Link	Hides the link to the resolve configuration error page and instead displays the error along with the constraint rule message.
Option Line Item Columns	Type the field name from the line item object displayed in the Option Hierarchy page. To add more than one field name separate the names by a new line or a comma.
Option Product Columns	Type the field name from the product object displayed in the Option Hierarchy page. To add more than one field name separate the names by a new line or a comma.
Percentage Field Precision	Type a number to specify decimal places for percentage precision. The default percentage precision is 2 decimal places.
Quantity Field Precision	Type a number to specify decimal places for quantity precision. The default quantity precision is 2 decimal places.
Resolve Configuration Page	Select the default Resolve Configure page. On Resolve Configuration Errors/Warnings, this page is used. • *ResolveConfig* • *ResolveConfigProducts*
Run Post Finalize Trigger In Async Mode	Indicates whether the finalization task should be run asynchronously.

Setting	Description
Save On All Actions	Saves the configuration when you click any action on the cart page.
Show Header	Displays the header tabs and the sidebar on the custom pages.
Enable Keyed Matrix Pricing	Select this if you have option products with large number of dimensions of type discrete. This also helps in reducing SOQL queries.
Skip Review	Skips the review step for items in the cart.
Static Criteria Fields	Type the field names whose value do not change after the records are created for a cart.
Show Tab View	Displays option groups as tabs instead of sections in the options page.
Enable Field Expressions	Enables you to use Expression Builder for attributes, option groups and options.
Pricing Batch Size	Type a number to define the number of line items that can be processed in a single pricing call. Setting the Pricing Batch Size, the system runs pricing with the specified number of products as a batch, thus increasing performance. These batch calls to the database are governed by the Salesforce CPU time limit and hence the number assigned for Pricing Batch Size must be carefully evaluated.
Pricing Profile	Type *Basic* or *Advanced*.A pricing profile is *Basic*, when there are none of the following used: • Pricing Rules • Price Matrices • Related Pricing Setup • BundlesNote: If the Pricing Profile field is left blank, the default value is *Advanced*.
	Hides the sidebar for the following Admin pages:

Setting	Description
Show Admin Sidebar	• CPQConsole.page • CategoryManager.page • ClassificationHierarchy.page • ConstraintRuleActionCriteriaEdit.page • ConstraintRuleConditionCriteriaEdit.page • CriteriaUpdate.page • DisplayActionSettings.page • DisplayColumnSettings.page • FeatureSet.page • HierarchyViewUpdate2.page • IconUploader.page • ManageGuidedSearchRule.page • MultipleConstraintRulesAdmin.page • PriceListItem.page • PriceRule.page • ProductConsole.page • ProductConstraintView.page • ProductDefaultRuleCriteriaEdit.page • ProductFilterMaintenance.page • RelateAttributeToProduct.page • RelateFeatureToProduct.page • RelateProductToBundle.page • RelateProductToCategory.page • RelateProductToFootnote.page • SystemProperties.page

Setting	Description
Custom Pricing fields	Type the custom field API name from the line item object displayed on the cart page. To add more than one field name, separate the names by a new line or a comma. When an end user makes any changes to these custom fields, the system is set to reprice before clicking Review and Finalize.
Use Button to Save Selection	Indicates whether to wait for button click before processing option selection and validation.
Use Enhanced CSS	Enables you to include the enhanced CSS file on CPQ pages.
Auto Update Category View	Indicates whether auto incremental updates to category view are enabled.
Cleanup Invalid Rule Prompt	Enable removal of line items prompted by currently invalid rules.
Compute Totals In Separate Step	Indicates whether the totaling should be performed in a separate step or it should be combined with the base pricing step. This setting ensures that the totaling, which is dependent on the number of lines, should be done as a separate remoting call. This setting helps to avoid the CPU time limit issue to a large extent.
Custom Deal Guidance Page	Name of the custom Deal Guidance page.
Custom Option Attribute Page	Name of the custom Option Attribute page.
Disable Pricing Query Optimization	

Setting	Description
	Indicator to disable pricing query optimization for cart. This setting allows the system to load large number of managed fields from the line item into the cache. By default, fewer fields are retrieved for line items in the cart to optimize the heap. This indicator is used to disable the optimization so that it supports the existing customization.
Enable Fast Cloning	Indicates whether fast cloning is enabled.
Enhanced Search URL	Search URL end point for enhanced product search.
Expand Bundles in Cart	Expands all attributes and options of a bundle product in the cart.
Field Expression Execution Mode	Indicates where the numeric expressions are evaluated. Following are the values: • Server - (Default) Enables the server side execution of the numeric expressions. • Client - Disables the server side execution of the numeric expressions.
GroupBy Fields	Specify the API names of those line item fields that are of the type: Lookup, Formula Fields, and Text. These line item fields are shown under the Group By drop down on the cart page. For example, for Group By PriceList, specify the API name of the field that is a lookup to the pricelist object.
LargeCartThreshold	The threshold of the number of line items in the cart, when exceeded indicates a large cart. If you specify -1 as a value, the system deactivates the threshold. A large cart influences how line items are processed in the application to stay under various platform governor limits. In this mode, the system performs Base Price and totalling calculations with modified behavior for the following checks:

Setting	Description
	• Deal Guidance and Approval checks are turned off. • Finalize and Reconfigure actions run certain operations, such as creation of line items, attributes, and usage tiers in Async mode.
Multi Currency Management	Contains *Platform* and *None* as values and is used to define the currency precision in a multi-currency org. • *None:* This is the default value. If you choose this value, the system considers the currecny precision settings that you have defined using Currency Field Precision. • *Platform:* If you choose this value, the system considers the precision settings for the currency defined at **Setup > Administer > Company Profile > Manage Currencies**. When you click **Manage Currencies**, the decimal places defined for your **Corporate** currency is obeyed. If you do not specify any value in the **Decimal Places**, the system considers the default value as 2.
Option Pricing Chunk Size	The option pricing chunk size for bundles with large number of options. The default value is 100. The SFDC governor limit affects the chunk size.
Product Attribute Extension Tables	Enter comma-separated API names of the lookup relationships, created with the **Product Attribute Value** object. For example, if you have 4 extension objects to **Product Attribute Value** object and their corresponding 4 lookup relationships, you must specify 4 API names in this field.
Product Display Max Length	Captures the maximum no of characters to be displayed for the product name on the cart page. If the product name exceeds the maximum lenght, the elipsis are shown. The default value is 21.
Run Misc Finalization Task in Async Mode	Indicator to check whether proposal line product attribute and usage tier records to be created with some delay after you **Finalize** a cart. In a scenario when the user creates Asset or generates a document immediately after finalizing the cart, there are chances that the record does not have attribute value or usage tiers.

Setting	Description
	If this checkbox is selected, the usage tiers and attribute record for proposal line items are created in Async mode when you finalize a cart. Otherwise, the usage tiers and attribute record for proposal line items are created as soon as you finalize a cart.
Show Attributes in Cart	Shows the specified attributes applicable to the line items in the cart. Attributes appear in the product dialog.
Term Field Precision	Type the precision value that you want to display for the **Selling Term** column on the shopping cart. For example, if your term calculation results to 4.553412 and you have set **Term Field Precision** to *3*, the **Selling Term** column displays 4.553. The default term precision is 5 decimals. This setting is applicable only for the New UI.
Revalidation Product Columns	Type the API names of product object fields that you want to display on the revalidation pop-up on the cart page. You can separate each field API name either by a comma or by a new line.

When a line item of **Charge Type** = *One Time* or a proposal that is *Evergreen* is created, you can choose to hide the dates of the line item or in the proposal. Similarly, if the Line Item has **Auto Renewal** marked as *true*, you can choose to hide the dates on those line items.

Scenario	Flags checked	Result
If **Price Type** is *One Time* on Line Item	Hide Dates For One Time (If Checked)	Hide **Start Date** and **End Date** on Line Item.
	Hide Dates For One Time (If Not Checked)	**Start Date** and **End Date** are visible.
If **Price Type** is *Recurring* or *Usage*	Hide Evergreen End Date (If Checked) and AutoRenewalType = Evergreen on Line Item	**End Date** is hidden.

Scenario	Flags checked	Result
	Hide Evergreen End Date (If Unchecked) and AutoRenewalType = Evergreen on Line Item	**End Date** should behave the same way it behaves for recurring.

Catalog Page Settings

This page lists all the custom settings you may require to set up the Catalog page. You can enter details in one or more of the following sections, based on your organizations requirement.

- General Page Settings
- Product List Settings
- Favorite Configuration Settings
- Mini Cart Settings
- Other Page Settings

General Page Settings

Setting	Description
Main Section Ratio	Type the ratio in which you want to divide the page. The ratio of the main sections in the page should be separated by a colon. The default value is 20:60:20. This is applicable to the *SelectConfigOptionsDetail*, *ConfigureBundle*, and *SelectBundleOptions* page.Note: Do not use the *SelectBundleOptions* page from your drop-down list, it is no longer supported.
Hide Default Options In Description	Hides the default options listed below the product description.
Hide Configure Action	Hides the Configure action for a Product on the catalog page.

Setting	Description
Read Only Location	Makes the Location field read-only.
Search Query Limit	Type a value to set the limit for a search query. For faster product search, type a smaller number. The default is 1000.
Hide Single Top Category	Hides the top category. This property is useful only when you have a single root category and want to have a cleaner look for all of its child categories, without the root category.
Hide Help Me Choose	Indicates whether to hide the Guided Selling link. By default this is not selected and the system shows Guided Selling link on the Catalog page for the respective categories.
Custom Action Label Name	Type the name of the custom label for the action.
Custom Action Page	Type the name of the custom action page.
Enable Category Filter	Filters the categories based on search results and displays in the Narrow Your Search area on the left.For example, if the search result has products from only two categories, the Narrow Your Search section displays only those two categories. You should always select **Enable Category Filter** when **Hide Narrow Your Search** is cleared.
	Enables caching of products of a price list containing less than 3000 products.

Setting	Description
Cache All Products (BETA)	
Show Checkbox In Rule Dialog	Enables check box to select multiple products in the Inclusion Constraint Rules dialog at once, instead of adding the products individually using Add to Cart. This setting works for *Inclusion Rule* with Min/Max Match Rule as *Include Min/Max*.

Product List Settings

Setting	Description
Listed Products Column2	(optional) Type the API name of the product field to display in the listed products section. Example Value: ProductCode
Listed Products Column3	(optional) Type the API name of the product field to display in the listed products section. Example Value: ProductCode
Hide Compare Products	Disables the Compare Products feature in the catalog page.
Hide Product Image	Hides the product image from the catalog page.
Show Quantity Input	Displays the Quantity field on the catalog page.
	Hides the price information column in listed product section. By default, this is not selected.

Setting	Description
Hide Listed Products Price Column	
Hide Breadcrumb	Hides the breadcrumb navigation element.
Carousel Default Icon	Type the ID of the icon. Ensure that you have attached the icon to a desired category using Category Hierarchy Manager, and **Large Icon Image** field is added to the standard layout of Category Hierarchy Detail page.
Hide Narrow Your Search	Hides the Narrow Your Search section that displays the category tree on the left of the catalog page. This setting is deprecated for the New UI.
Disable Add To Cart for Option Product	Hides the **Add To Cart** and **Configure** buttons on the Catalog page for an option product. This is helpful to prevent your sales representative from selling an option product without its bundle product.

Favorite Configuration Settings

Setting	Description
Disable Favorites	Disables the Favorites category on the catalog, **Save As My Fav** button on the cart and any UI elements related to Favorite Configurations on catalog, configuration and cart pages. By default, this checkbox is not selected.
Listed Favorite Configurations Column 1	Enter the API name of the favorite configuration field that you want to display on the catalog page for your favorite configuration records. You can enter a maximum of one API names. If you have a requirement of displaying one more field, use **Listed Favorite Configurations Column 2** custom setting.

Setting	Description
Listed Favorite Configurations Column 2	Enter the API name of the favorite configuration field that you want to display on the catalog page for your favorite configuration records. You can enter a maximum of one API name.
Favorites Display Label	Enter a name for the group of favorite configuration records. This label is displayed as a category name on the catalog page. If you do not specify any name, the system gives *Favorites* as the default name.
Favorites Upload Image	You can choose an image to upload for your group of favorite configuration category. This image is displayed under your favorites (as entered in **Favorites Display Label**) category on the catalog page. By default, no image is set.
Save as Favorite Dialog Columns	Enter comma-separated API names of the columns that you want to display on the **Save As My Favorite** dialog box on the cart.

Mini Cart Settings

Setting	Description
Selected Products Column2	(optional) Type the API name of the line item field to display in the selected products section. Example Value: ProductId__r.ProductCode
Selected Products Column3	(optional) Type the API name of the line item field to display in the selected products section. Example Value: ProductId__r.ProductCode
Selected Products Column4	(optional) Type the API name of the line item field to display in the selected products section. Example Value: ProductId__r.ProductCode

Setting	Description
Order Status Fields	Type the name of the line item fields (separated by comma) that contains information about promotion, delivery info, shipping info, and more.
Hide Selected Products Column1	Hides the column that displays the product name.
Minimize Cart	Displays the cart icon and hides the cart sidebar on the right hand side of the catalog page.
Show Selected Product All Charges	Displays all the charge types for selected products in the tree view options page.

Other Page Settings

Setting	Description
Show Product Icon Cart Detail View	Displays the product icon on each line item in the Cart Detail View page. Note: This will disable the action item columns and show the actions under the product icon.
Show Selected Products in Config Options	Displays the Selected Products widget in the Config Options Detail View page.
Collapse All Leaf Option Groups	Collapses all the leaf option groups in the Config Options Detail View page.
Show Recommended Products Cart View	Displays the recommended products component in the Cart Detail View page.

Setting	Description
Cart List Item Description Field	Type the API field name of the Long Description field to be used in the Cart Detail View page.
Hide Line Item Attribute Details	Hides the Line Item Details section on the Product Attribute Detail page.

Bundle Page Settings

This page lists all the custom settings you may require to set up the Options page. You can enter details in one or more of the following sections, based on your organizations requirement.

- General Page Settings
- Option List Settings
- Mini Cart Settings

General Page Settings

Setting	Description
Main Section Ratio	Divides the options page with the specified ratio. The ratio of the main sections in the page should be separated by a colon. The default value is 0:70:30. This is applicable to the *SelectConfigOptionsDetail*, *ConfigureBundle*, and *SelectBundleOptions page*.Note: Do not use the *SelectBundleOptions* page from your drop-down list; it is no longer supported.
Show Tab View	Displays option groups as tabs instead of sections in the options page.
Bundle Description Field	Type the field names from the product object to be displayed as bundle description.
	Indicates whether to wait for a button click before processing option selection and validation.

Setting	Description
Use Button to Save Selection	

Option List Settings

Setting	Description
Listed Options Column 2	Type the API name of the product field to display in the listed options section.
Listed Options Column 3	Type the API name of the product field to display in the listed options section.
Hide Price Column	Disables the price column from the options page.
Show Option Quantity	Displays the Quantity in the body of the options pages. This setting controls the SelectConfigOptionsDetailView and ConfigureBundle page.
Hide Disabled Options	Hide the options that have been disabled by the exclusion rules. Such options will not be shown in the list of available options. Also, if all such disable options are under one option group, this group is also hidden on the configuration page.
Max Inline Option Attributes	Specify the numeric value for the maximum number of inline option attributes to be displayed in the attribute group. The default value is 3.

Mini Cart Settings

Setting	Description
Show Bundle Detail	Shows the bundle details in the options page.

Flow Settings

This page contains the settings to determine the custom or out-of-the-box pages when an end user clicks **Configure Products** button. Once setup, these flow settings override the Visualforce pages setup in custom settings.

You can enter details in one or more of the following sections, based on your organization's requirement.

Setting	Description
Name	Type a mandatory name for the flow.
Assets Page	Type the Visualforce page name for assets. Valid values are: • SelectInstalledProducts • SelectInstalledProductsEn
Cart Page	Type the Visualforce page name for cart. Valid values are: • Apttus_Config2__CartDetailView • Apttus_Config2__CartView
Catalog Page	Type the Visualforce page name for product selection. Valid values are: • Apttus_Config2__SelectConfigProducts • Apttus_Config2__SelectConfigProductsFilterView
Options Page	Type the options page to capture the options for a bundle product. Valid values are: • Apttus_Config2__ConfigureBundle • Apttus_Config2__SelectConfigOptions

Setting	Description
	• Apttus_Config2__SelectConfigOptionsTabView • Apttus_Config2__SelectConfigOptionsListView • Apttus_Config2__SelectConfigOptionsTreeView • Apttus_Config2__SelectConfigOptionsDetailView • Apttus_Config2__SelectBundleOptionsNote: Do not use the *SelectBundleOptions* page from your drop-down list, it is no longer supported.
Product Attribute Detail Page	Type the Visualforce page name to capture product attribute detail information. Valid value is: • Apttus_Config2__ProductAttributeDetail3

Installed Products Settings

This page lists all the custom settings you may require to set up the Installed Products page. You can enter details in one or more of the following sections, based on your organizations requirement.

Setting	Description
Direct Configure Asset Actions	Type comma-separated values of Increment, Amend, Renew, Cancel actions. Listed actions will configure products directly. This setting is no longer supported. You can configure the Asset Actions at each Asset Line Item.
Max Renews Per Trip	Type the maximum number of renewals per trip. To prevent CPU timeout use a smaller number. The default is 40.
Amend Change Fields	Type comma separated line item field names whose values can turn an existing asset line into an amended asset line. This setting is no longer used. All the editable fields honor the flow settings configured for the Cart.

Setting	Description
Allow Mass Change	You can enable or disable mass changes for *must configure* assets with the help of this setting.
Default Renewal Term	Type a value for a renewal term. When the value is provided use this value as the renewal term for Renewing products. The value here denotes months to renew as the end date. Ideally, your **Selling Term** is equal to your **Default Renewal Term**.
Allow Backdated Termination	Enables you to terminate an asset by specifying an end date to any date prior to the current date. End date should be greater than the start date. Backdated Termination is supported out-of-the-box.
Hide Asset Actions	Type comma-separated values of Increment, Amend, Renew, Cancel actions. Listed actions will be hidden from the users. Currently, this setting is not supported. You can configure the Asset Actions at each Asset Line Item.
Show Accounts Filter	Selecting this displays the account filter in the installed products page. This is currently not supported.
Filter Fields	Type the *API name* of Asset Line Fields to be displayed on the **Filter By** panel. Works with text fields, Picklist, Multi-picklist, Date, DateTime and Boolean.
Hide Co-Term	Hides the *Co-Terminate End Date to* field with its values, the text field to enter date, and the *Apply* button. This setting is applicable to the classic UI only.
Enable One Time Change	Select this flag to enable modifications or amendment to one-time assets. This is a global setting and cannot be restricted to apply for specific products. You can perform only amendments on one-time assets and not renewals. If you deselect this and modify the asset, the renewal quote is updated for the modified asset. If you set it to *false*, Renewal Opportunity and Renewal Quote is not created for the one-time only products, but if you modify the asset, the change is reflected in the Renewal Quote. If you set it to *true*, Renewal Opportunity and Renewal Quote is not created for one time.

Setting	Description
	It is recommended that you treat one-time assets purely as one-time sales. Any modifications or renewals should not be encouraged for one-time assets.
Renew One Ramp	Consider a scenario where multiple ramped assets exist for a quote. In the previous releases, renewing any one of the ramped assets results in the creation of a renewal cart with the same number of ramps as the original proposal . Each ramp line is created from the original deal which implies that the uplift has to be defined from Ramp 1 for accurate prices in the subsequent ramps. Redefining and specifying details for each ramp can be cumbersome and as a user, you would expect that for a three year ramped deal, the uplift specified on year 3 is applied on renewal to year 4. This setting enables you to renew only one ramp line item based on the new term in order of renewal term in the asset line item, default renewal term specified in the Installed Product Setting and Selling Term defined in the asset line item.

Proposal System Properties

This page lists all the Quote/Proposal system properties. You can enter one or more of the following property details.

Setting	Description
Admin User	The admin user is the default owner of activities created by a user who is not allowed to be the owner (for e.g. customer portal user). Enter the admin user name as first name, last name.
Auto Create Order	Select this check box if you want to create order and asset as soon as a Quote /Proposal is accepted.
Bypass Sharing	Indicates whether apex code can bypass record sharing during selective operations such as clone and deleting draft attachments.

Setting	Description
Default Opportunity Quote Owner	Type *Opportunity Owner* or *Current User* to set the default owner for the quote /proposal created from an opportunity. If not set, *Opportunity Owner* becomes the default owner.
Default Quote Agreement Owner	Type *Quote Owner* or *Current User* to set the default owner for the agreement created from a quote/proposal. If not set, *Quote Owner* becomes the default owner of the new agreement.
Email Template For Presenting Proposals	Type the name of the email template to use for presenting proposals.
Enable PDF Security	Enabling PDF security lets users apply security settings to PDF documents and protect them with a password.
Instance URL	Type the Salesforce instance URL. For example: https://na7.salesforce.com This is required to navigate to custom pages in the managed package.
Merge Call Timeout Millis	Type a number to indicate timeout in milliseconds for the merge request. For example: 60,000
Merge Webservice Endpoint	Type the Apttus merge webservice endpoint. For example: https://mwsdev.apttus.net/cgi-bin/360/MergeServer/Bin/MMCGI.exe
PDF Owner Password	Type the password required to change permissions of the PDF document like printing or editing.

Setting	Description
Auto Sync With Opportunity	Automatically synchronizes the quote line items with the opportunity, when the proposal is finalized. This setting appears only if you have Apttus Quote/Proposal-Configuration Integration package installed.
Disable Opportunity Products Copy	Disables copying of opportunity products to the quote/proposal.
Sync Bundle Using Line Items	Synchronizes bundle products using proposal line items. The default uses proposal summary objects. This setting appears only if you have Apttus Quote /Proposal-Configuration Integration package installed.
Sync Option Products	Synchronizes options along with bundles when you click **Sync with Opportunity** for your proposal. The default is synchronize bundle only. This setting appears only if you have Apttus Quote/Proposal-Configuration Integration package installed.
Max Child Level	Type the maximum level to generate the merge data for the proposal.

Comply System Properties

This page lists all the Comply system properties. You can enter one or more of the following property details.

Setting	Description
Admin User	The admin user is the default owner of activities created by a user who is not allowed to be the owner (for example, customer portal user). Enter the admin user name as first name, last name.

Setting	Description
Auto Enable PDF For Final Docs	Select this and the **Create PDF Attachment** check box is always selected when you choose to save as *Final - to be signed* from the check-in dialog. You may use this field when you want to finalize an agreement document.
Auto Enable Reconciliation	Select this and the *Reconcile Document* option is always selected when you go to check-in an agreement document.
Agreement Number Field For Imported Docs	In this field, specify the API name of the field you want to use. For example, useApttus__Agreement_Number__c. When a new document is imported into the system, it will include the agreement number in the top right corner of the header on each page, using the field selected above. For more information about when to use this field, refer Agreement Number/Header Configuration.
Allow PDF Select Override	This is only applicable when **Auto Enable PDF For Final Docs** is selected.
Allow Reconcile Selection Override	This is only applicable when **Auto Enable Reconciliation** is selected.
Auto Create Order	Select this check box if you want to create an order and asset as soon as an agreement is activated. This is useful when you want to integrate Assets with Contracts.
Auto Enable Private Indicator	Select this check box if you want to auto enable private indicator for documents. If you select this check box, the **Make this document private** check box for any agreement document is auto selected.
Auto Insert Header Footer Data	Select this field to automatically insert the Agreement Number Field For Imported Docs field value to the header and the latest timestamp to the footer of an agreement document. This field is available for Generate, Import and Offline actions. For more information about when to use this field, refer Agreement Number/Header Configuration.

Allow Private Selection Override	Select this check box if you want to allow the user to override the private document selection. This setting is applicable only when **Auto Enable Private Indicator** is selected for documents.
Auto Sync With Opportunity	Indicates whether the agreement will be automatically synchronized with the opportunity when the agreement is accepted. This setting appears only if you have Apttus Contract-Configuration Integration package installed.
Enable PDF Security	Enabling PDF security lets users apply security settings to PDF documents and protect them with a password.
Bypass Sharing	Indicates whether apex code can bypass record sharing during selective operations such as clone and deleting draft attachments.
Contract Summary Template	This field contains the name of the contract summary template. You may have to define a separate template to contain the contract summary details and mention the name of the template here.
Default Opportunity Agreement Owner	Type *Opportunity Owner* or *Current User* to set the default owner for the quote /proposal created from an opportunity. If not set, *Opportunity Owner* becomes the default owner.
Default Document Tags	Enter a comma-separated list of tags to make available for any agreement document as it is checked in from X-Author Contracts. Use of these tags requires Contract Document Versioning to be enabled. The **Document Finder** feature must also be configured in your org. For more information, refer to Configuring Document Finder.
Document Naming Convention	Specify a value to apply a custom naming convention for all agreement documents at generation, check-in and signature events. The default naming convention is %:Name%_%action%_%templatename%_%timestamp% If this property is left blank, the system will use the default naming convention.

Setting	Description
	The following attributes permitted when formulating a document naming convention are: • %checkintype% • %action% • %templatename% • %user% • %timestamp% • Agreement attributes such as %:Name%. Note: any variable prefixed by ':' represents a field on the Agreement object. Example agreement document name using the default naming convention: *SOW_Regenerated_SOW ABC_2015-08-07*
Document Structure FX2 For Imported Docs	Check this box to make document structure FX2 format for all Offline documents (created or imported). If not checked, all offline documents are created in the "pre-existing" format. This property only applies to Offline agreements created from the Agreement or user Home page links in Salesforce. The format of Offline agreements created using X-Author for Contracts will still depend on user-input from X-Author.
Email Template For Checkin Notification	Type the email template for sending check-in notifications.
Enable Clause Approvals	This check box allows you to set up approval processes on clauses used in your document.

Setting	Description
Enable Document Versioning	Check this box to enable Document Versioning. ***All new agreement records created in your org will use Document Versioning after this setting is enabled***. Enable Version Control must be enabled for Document Versioning to work properly. Enabling Document Versioning changes the value of the **Version Aware** Agreement field to TRUE for all new agreements after the property is activated. The **Version Aware** field is a flag that tells Apttus Contract Management to use Document Versioning for a specific record. **Important Note**: It is recommended that once a record is flagged as Version Aware, you do not disable this field, as versioning will become undefined for the agreement record in question. Instead, ensure that records which should not use Document Versioning do not have the field enabled when they are created. For more information on how you can configure Document Versioning for your org, refer to Enabling Contract Document Versioning.
Enable Merge Call Debug	Enable Merge Service debugging.
Enable Multiple Checkout	Allows multiple checkout. This is applicable only when version control is in effect.
Enable PDF Security	Enabling PDF security lets users apply security settings to PDF documents and protect them with a password. See Enabling PDF Security for Quote/Proposal Documents for more information.
Enable Submit Merge Call	Submits merge calls for processing. A Submit button is displayed during document generation.
Enable Template Versioning	Check this box to enable Template Versioning. *All new Templates created in your org will use Template Versioning after this setting is enabled.* For more information, refer to Template Versioning.

Setting	Description
Enable Version Control	Check this box to enforce a check-in/check-out policy for agreement documents. This setting must be enabled when "Enable Document Versioning" is checked. Note: This property existed in CLM versions prior to 8, so this property may already be enabled.
Footer Datetime Format For Imported Docs	Specify the format in which date and time will be shown in the generated agreement. When a new document is imported into the system, it will include the Date in the bottom left corner of the footer on each page, in the format selected above. To know the supported date and time formats for footer, refer Setting date and time format for footer.
Instance URL	The Salesforce instance url for redirecting to custom pages. For example: https://na7.salesforce.com This is required to navigate to custom pages in the managed package.
Max Child Level	The maximum level of lookups and child objects available from the parent object to generate the merge data for. For example, **Agreement** is the parent object, **Agreement Fee** is the child object of **Agreement** and **Agreement Fee Adjustment** is the child of **Agreement Fee** and grandchild of **Agreement**.
Merge Call Timeout Millis	The timeout in milliseconds for the merge request. Type a number to indicate timeout in milliseconds for the merge request. For example: 60,000
Merge Webservice Endpoint	Type the Apttus merge webservice endpoint. For example: https://mwsdev.apttus.net/cgi-bin/360/MergeServer/Bin/MMCGI.exe
PDF Owner Password	Type the password required to change permissions of the PDF document like printing or editing.
Publish Author Events	Indicates whether merge events that occur in X-Author, such as check-in or check-out, should be published. If enabled, a record is inserted in the Merge Event table on the Agreement record.

Setting	Description
Publish Merge Events	Indicates whether merge events, such as generating an agreement document or creating offline document should be published. If enabled, a record is inserted in the Merge Event table on the Agreement record.
Sync Bundle Using Line Items	Synchronizes bundle products using agreement line items. The default uses agreement summary objects. This setting appears only if you have the Apttus Contract-Configuration Integration package installed.
Sync Option Products	Synchronizes options along with bundles. The default is synchronize bundle only. This setting appears only if you have the Apttus Contract-Configuration Integration package installed.
Temp Email Template Inactive Hours	Use this property to set the orphan period of temporary email templates to use in conjunction with the **CleanupJobScheduler** APEX class. The default value of this property is set to 4 hours. For more information, refer to Temporary Email Template Cleanup.
Use Agreement Locks for Versioning	Check this box to use Agreement locks for versioning instead of document-level locking: • If enabled, all agreement documents are locked (by the user checking out) when ***any one of them*** is checked out. • If disabled, only checked out documents are locked (by the user checking out)–any other agreement documents which have not yet been checked out can be checked out for editing.

Lookup Field Settings

Lookup field settings provide an ability to filter out look up records through filter criteria based on immediate parent on which the lookup field is defined. Each new lookup field shown in the AngularJS cart needs a corresponding lookup field settings record.

Before you set up Lookup field settings, ensure the following setup is configured.

Save Bundle Page Settings

Save the default values of the bundle page as is.

1. Navigate to **Config Settings > Bundle Page Settings**.
2. Verify the settings defined for the bundle page.
3. Click **Save.**

Following are the scenarios where lookup field settings are required:

- ### Make your product attribute as hidden

  While configuring a product, along with its attributes, the product attribute lookup will now display only relevant attributes for the products which you have just selected on the catalog page.

  Perform the following steps to hide the Product ID attribute.
  Prerequisite: You must create a custom formula field, named *Product Id* to fetch the *Product Id* of the Line Item object.

  1. Navigate to **Product Attribute Groups** by clicking **All Tabs**.
  2. Select the appropriate Product Attribute Groups and click **Edit**.
  3. Select the **Is Hidden** check box for Product Id.
  4. Click **Save**.

To customize the lookup fields, create a new record under **Custom Settings > Lookup Field Settings > Manage** and fill in the requisite details.

Field	Description
Name	Specify the name for the custom setting.
Display Columns	Specify the API names of fields to be displayed on the lookup dialog box, separated by a comma.
Filter Criteria	Enter the valid expression to further filter down the search results for a lookup. Example: Product__c=Apttus_Config2__ProductAttributeValue__c.Product_Id__c This will fetch only such attributes whose IDs match with the selected products. Here, the ProductId is a custom formula field which fetches the Product Id of the Line Item object.

Field	Description
Lookup Field Name	Specify the API name for the lookup field. Example: Product_Edition__c This is the lookup field containing values such as *Enterprise*, *Standard*, and *Premium*.
Object Name	Specify the object name from which the lookup relationship is created. Example: Apttus_Config2__ProductAttributeValue__c

- ## Configuring Users and Queues for Quote Collaboration

You need to configure lookup fields for users and queues so that Quote Collaboration pop up shows the relevant users and queues while assigning the collaboration request.

For configuring User, ensure that you set up the lookup field (under **Custom Settings > Lookup Field Settings > Manage**) with the following values:

Field	Description
Name	`CollaborationRequest__c.User.OwnerId`
Display Columns	`Name`
Filter Criteria	`UserType = 'Standard' AND IsActive = true`
Lookup Field Name	`OwnerId.User`
Object Name	`Apttus_Config2__CollaborationRequest__c`

For configuring Queue, ensure that you set up the lookup field with the following values:

Field	Description
Name	`CollaborationRequest__c.Group.OwnerId`
Display Columns	`Name`
Filter Criteria	`Type = 'Queue'`
Lookup Field Name	`OwnerId.Group`
Object Name	`Apttus_Config2__CollaborationRequest__c`

Object Summary Settings

You can use the **Object Summary Settings** to display the summary information for the Proposal (cart header) and Product in the Salesforce.

Navigate to **Setup > Develop > Custom Settings > Object Summary Settings.**
Click **Manage** for Object Summary Settings, and then click **New**.

In the **Object Summary Settings Information** section, provide the following information:

For Proposal Summary

Field Name	Description
Name	Specify the name as: Proposal Summary
Object Name	Specify the object name as: *Apttus_Proposal__Proposal__c*

Field Name	Description
Display Fields	Specify the API names as: *Name, Apttus_Proposal__Proposal_Name__c, Apttus_Proposal__Account__c, Apttus_QPConfig__PriceListId__c*

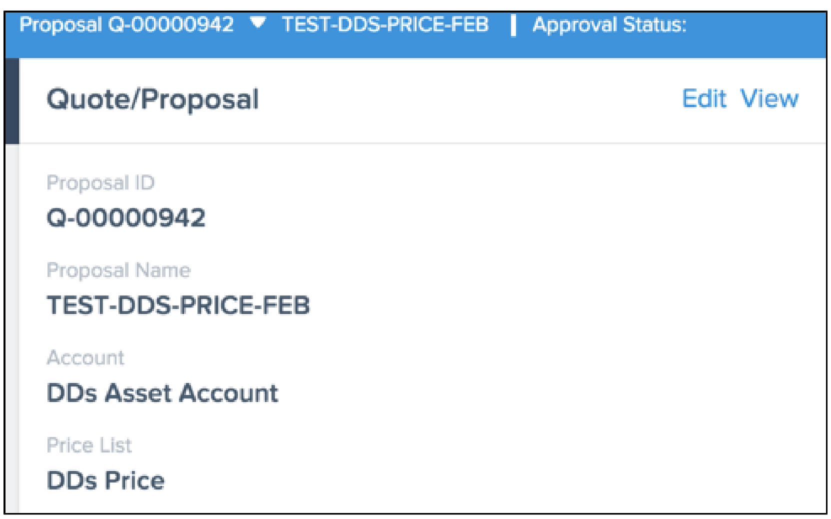
You can specify any Quote/Proposal field in a comma-separated list.

For Product Summary

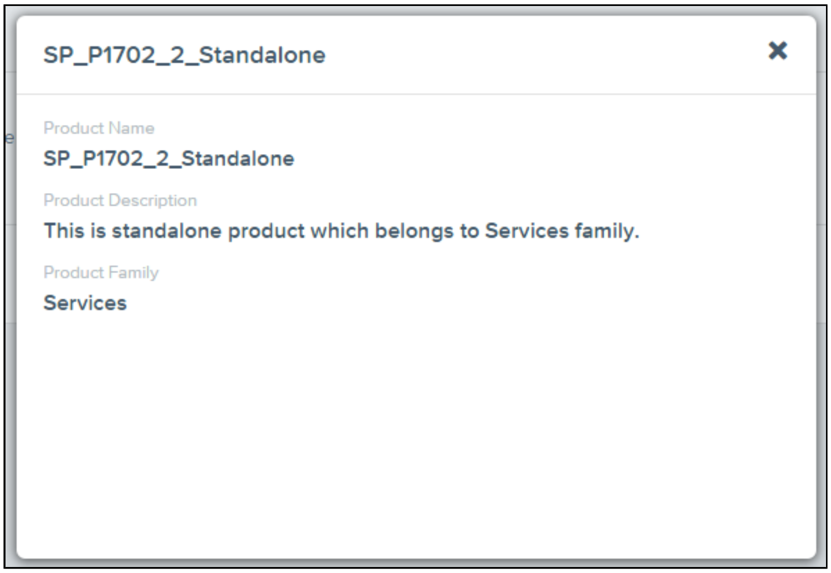
Field Name	Description
Name	Specify the name as: Product Summary
Object Name	Specify the object name as: *Product2*
Display Fields	Specify the API names as: *Name, Description, Family*

You can specify any Product2 field in a comma-separated list.

On the catalog page, when you click **Proposal ID** on the top left corner, you see proposal summary with the fields configured in Proposal Summary.



Similarly, when you click **Product Name** on the catalog page, you see product summary page with the fields configured in Product Summary.



If there is no **Object Summary Setting** defined in the system, only Name will be displayed by default when you open the Proposal or Product Summary dialog box.

For Favorite Configurations

Field Name	Description
Name	Specify the name as: *Favorite Configuration*
Object Name	Specify the object name as: *Apttus_Config2__FavoriteConfiguration__c*
Display Fields	Specify the API names as: *Name, Apttus_Config2__PriceListId__c, Apttus_Config2__ConfigurationId__c*

Smart Search Settings

As Apttus customers look to tackle the large sizes and the complexity of their product catalogs, with multiple levels of categories and stricter rules, the need for smarter search is more important than ever. The new smart search provides much faster results, type-ahead mechanism, and faceted search to help narrow the search results and still honor product visibility and availability rules.

Please contact Apttus Product Support team to enable the Smart Search capability for your org.

- Smart Search Config Settings
- Setup Apttus Intelligent Cloud Search Connection Settings
- Add and Review Product Field Weightage
- Push Data into Search Engine
- Refresh Data in Search Engine
- Remove Data in Search Engine
- Review Sync Status
- Switch to Azure Smart Search

Smart Search Config Settings

Go to **Config Settings > Smart Search Settings**. This page lists all the smart search specific settings that you need to set up.

Setup Apttus Intelligent Cloud Search Connection Settings

Please contact Apttus Product Support/PS/CSM team to obtain the URL and API User Key.

Setting	Description
Enhanced Search URL	Specify the URL end point for enhanced product search. With **Enhanced Search URL** populated with the correct URL, the system synchronizes all the changes at every hour and displays the results in the **Review Sync Status** section. If you do not specify the **Enhanced Search URL**, the system neither performs the sync nor updates the information in the **Review Sync Status** section .
API User Key	Specify the API key that consists of the client ID and is used for smart search related activities. This key is required before Search Activation/Deactivation and resync. An error is displayed if you activate, deactivate or resync without the API User Key.

Click **Save.**

Add and Review Product Field Weightage

The weightage allocation enables the Smart Search to control the ranking of the search results based on customer needs. The total for the weightage is 100% and this can be split into any number of Product Fields that participate in the search criteria. For example, 70% weightage to product name and 30% weightage to product code means that if for a search text there is a match with a product name as well as a product code, the product name match will be ranked higher in the result. This capability allows the customers to fine tune the search results and also include custom fields that may be included in the search criteria.

To define the criteria, the admin is required to define the map as follows. The map of fields and corresponding weightage can be different for each flow.

1. Select the desired flow using **Flow** drop down.
2. Select the desired field using **Field Name** drop down. Select the desired weightage in percentage using **Weightage** drop down.
3. Click **Add Row**, if you need to add an additional field. You can add product fields such as Product Name, Description, Product Code, and more. Ensure that total weightage sum of all the fields equals to 100%.
   The system supports fields of type text, picklist, rich text, and number. It can also be a mix of all these types of fields. You can add a maximum of 6 fields.
4. Click **Save**.

Push Data into Search Engine

This is a one-time Activation step. This enables the system to set up new indexes and push the product data that runs the catalog and search.

Click **Search Activation.**

Refresh Data in Search Engine

This step is required to be executed any time there is change to the base line setup or “search meta-data”. This includes:

- New fields are added to the weightage criteria.
- New fields are added to the Refine Search filter.

- New fields are added to the product catalog listings.

Click **Search Metadata Refresh.** You need to refresh the metadata only when you have made changes to the product schema or data.

Remove Data in Search Engine

This step is required when you want to deactivate the Smart Search functionality in your org and purge data from our servers.

Click **Deactivate Search**.

Review Sync Status

This section displays the details about the last Sync job performed for Smart Search Settings. The fields such as **Sync Message**, **Last Attempted Sync**, **Last Successful Sync** and **Current Sync Status** depict the necessary information about the Sync job. The Review Sync Status is changed when you activate Smart Search (by clicking **Search Activation**), run the batch job and deactivate the Smart Search (by clicking **Deactivate Search**).

Switch to Azure Smart Search

You can now switch the search engine from *Solr* to *Azure* for faster search results. Create a new record under **App Setup > Develop > Custom Settings > Config Smart Search Settings > Manage** with name *SmartSearchSync.* Inside this record, specify the **Search Environment** as *Azure.*

> ⚠ **Note**
>
> By default, the value in **Search Environment** is set to *Azure* for new package installation. For upgrade (if you are already using Smart Search), you must deactivate the smart search, change the search environment and then activate the smart search.

Application Settings Management

The Application Settings Management feature allows you to configure and enable pages on the CPQ Admin console to easily configure different settings of the CPQ application from one single URL.

- **Launch from new Admin UI:** Settings can simply be modified by navigating to a new section in the New CPQ Admin user interface, allowing a logical grouping of the settings.

- **Logical grouping of settings**: By default, all settings related to pricing will be grouped together and items related to the Configuration Engine are grouped together.
- **Admin configurable settings:** You can also create your own group settings for customer specific settings that are not part of the standard Apttus CPQ settings.

Adding the Application Settings Management Objects as Tabs

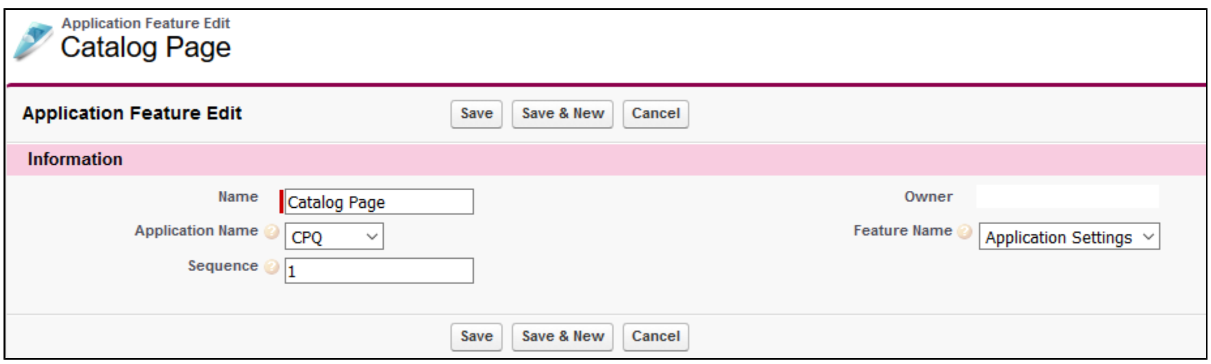
Go to **Create > Tabs** and create tabs for the following Objects.

Object	Description
Application Feature	This tab is the parent tab, all the application settings available under Config Settings, Quote/Proposal System Properties can be grouped under Application Features. For example, create an Application feature called as Catalog. You can group all the settings that affect the Catalog page under this application feature.
Setting Group	This should be a related list in the Application Feature Visualforce page. Using this page, you create logical groupings for each of the settings as well as the Flow in which these settings will be available in. For example, for the Catalog page, you want to group settings by Display Settings and Navigation and Actions. The Display Setting group can have all the application settings that control the components displayed on the Catalog page, such as Hide Compare Products, Show Quantity. The Navigation and Actions Settings group will have all the action and navigation settings such as, redirect to another page on button click without showing the spinner by using the Quick Redirect Action type.
Application Setting	This should be a related list in the Setting Group Visualforce page. Each Application Setting has to be defined based on its behavior within a setting group. For example, you can group all application settings controlling the display on the Catalog page together. For example, the Show Quantity application setting will be a part of the Display Setting group.

Creating an Application Feature

When you navigate to the Application Feature tab and click **New** to add and save details in the fields. You can create multiple application features.

Field	Description
Name	Name that you want to give to the application feature. Let's name the Application feature as *Catalog Page*.
Application Name	The Application in which this feature setting will be applicable. By default the value is CPQ. If you want to add additional values, navigate to the Application Settings object and add additional fields to the Application Name picklist such as CLM.
Feature Name	The feature for which this Application Setting is defined. For example, Cart Page, Catalog Page, Flow Setting, Proposal Settings. You can add values to this picklist by navigating to the Application Settings object and add additional fields to the Application Name picklist. An example of a feature name could be Catalog page, Cart page, or just Application Settings.
Sequence	The sequence in which the Application Feature appears.

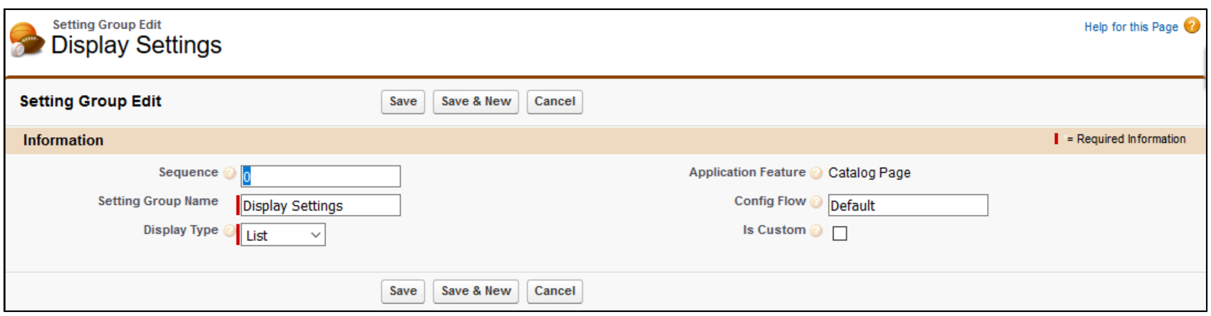


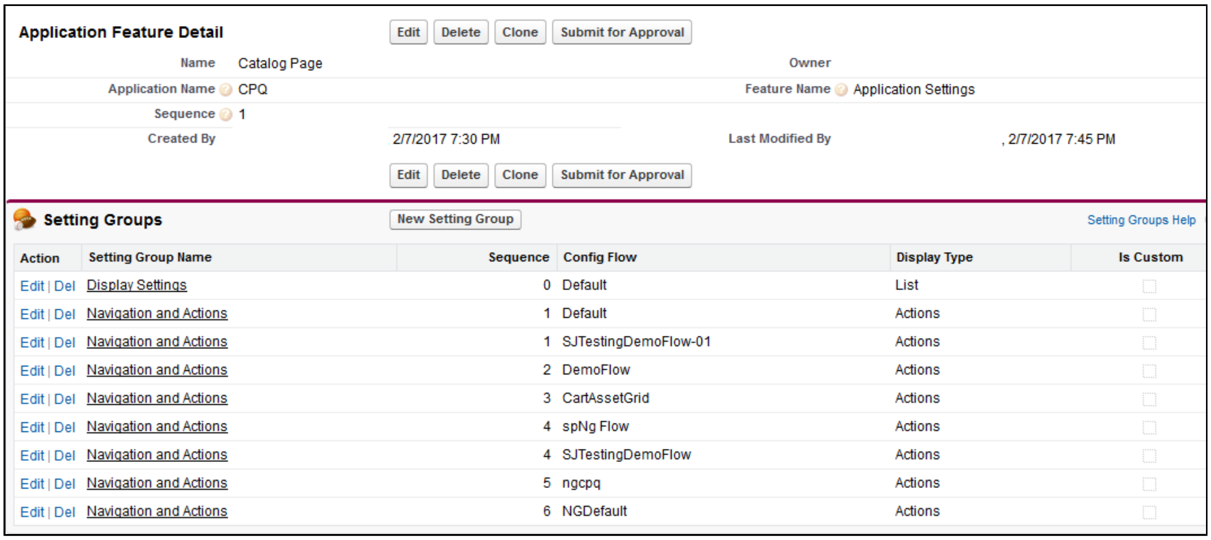
Creating a Setting Group

After you have created an application feature, use the Setting groups related list from the Application Feature Visual force page to create a new Setting Group. You can create multiple Setting Groups within an Application Feature. Click New to add and save details in the following fields:

Field	Description
Sequence	The Sequence in which the Setting group will appear on the New CPQ Admin Page.

Field	Description
Setting Group Name	The name you want to give to the Setting Group, such as Display Actions.
Config Flow	Specify the name of the flow in which the setting group is applicable and visible in.
Display Type	Specifies how the application settings defined in the Setting Group appear. Set the display type on the group based on the Application setting you define in the group. For example, all System settings would have display type as Lists, if all the application settings in the group control or specify custom display actions, the display type would be Actions and so on.
Is Custom	Specifies whether the Setting Group is a custom defined setting group.





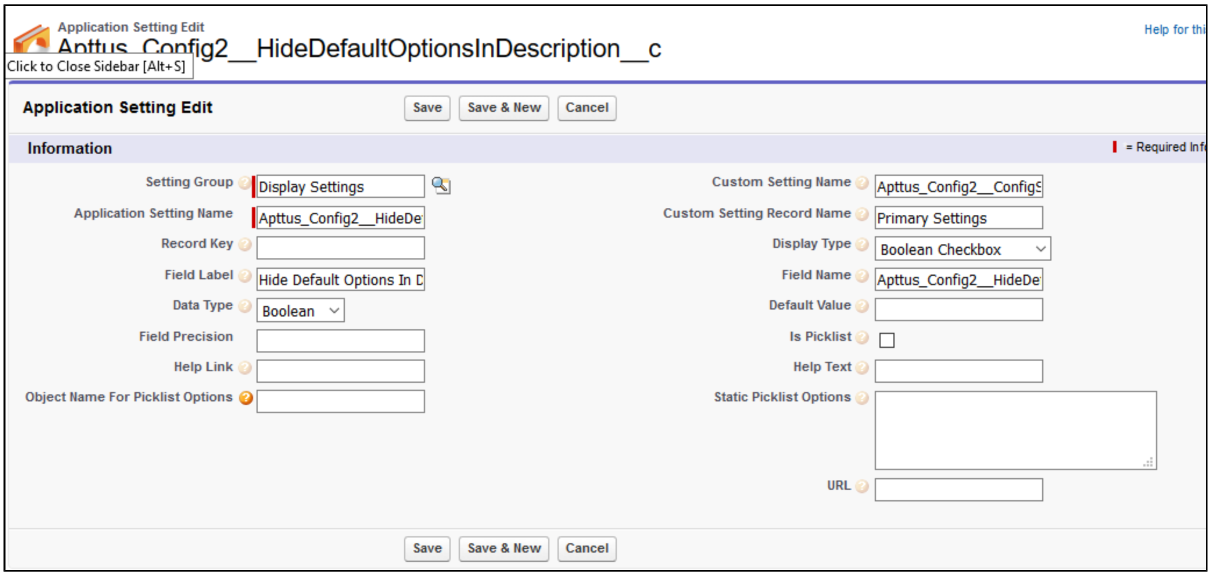
Creating an Application Setting

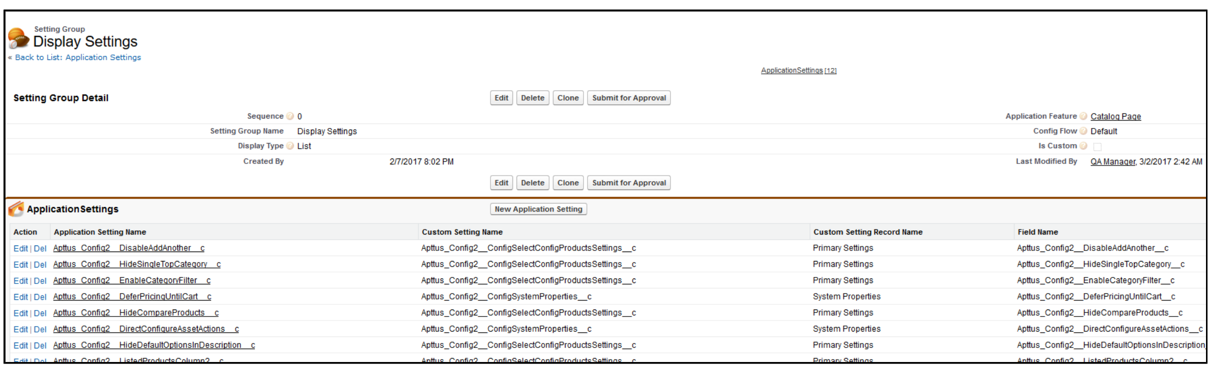
After you have created a Setting Group, use the Application Features related list from the Setting Groups Visualforce page to create a new Application Setting. You can create multiple Application Settings within a Setting Group. Click **New** to add and save details in the following fields:

Field	Description
Setting Group	A lookup to the setting group the application setting will belong to.
Application Setting Name	Specify a name for the application setting.
Record Key	Records associated with the custom setting will be fetched using a record key. When your setting Group is of type Actions or Columns, specify a record key. Record Key is used to identify record(s) in the associated Custom Setting.
Field Label	Name of the application setting. For example, Hide Default Options In Description.
Data Type	This value can be Boolean, number, picklist, text. In our use case, Hide Default Options in Description is a checkbox so the data type will be boolean.

Field	Description
Field Precision	This field specifies the to specify the number of decimal places shown for currency, quantity, and percentage precision fields.
Object Name For Picklist Options	Specify the API Name of an object whose fields are displayed as available options if the setting is a picklist.
Custom Setting Name	Specify the API name of the custom setting which contains the application setting. For example, the Hide Default Options in Description Setting is available in General Page Settings of the Catalog Page which is essentially defined as Apttus_Config2__ConfigSelectConfigProductsSettings__c. So you'll pass the API name of the Setting containing the Application Setting.
Custom Setting Record Name	Name of the custom setting to which the application setting belongs to. For example, Hide Default Options In Description is a part of the Primary Setting. Specify the Record Name as Primary Setting. The values can be System Properties, Primary Settings, Config Custom Classes and so on.
Display Type	Defines how the Application setting is displayed in the CPQ Admin Console page. This value can be Boolean, Multiselect picklist, picklist, text, or radio button. In our use case, Hide Default Options in Description is a checkbox so the data type will be boolean.
Field Name	API name of the custom setting. For example, Apttus_Config2__HideDefaultOptionsInDescription__c.
Default Value	Specify a default value of the setting. For example for Hide Default Options, in Description field, the value can be set to *true*.
isPicklist	Is a boolean checkbox specifying whether the application setting is a picklist.
Static Picklist Options	Specify a comma separated list of the picklist options if the application setting is a picklist.

URL	Specify a URL which provides more information for the custom setting.





All the settings once configured are available in the New CPQ Admin Console. The users can go and change each setting using the System Settings Page.

New UI

Apttus CPQ has an enhanced user interface based on an AngularJS framework. You can use the Apttus configuration and pricing engines on a web based, cloud-enabled, single page application that can operate at massive scale and real-time speed.

The following features pertain to the New UI:

- Configuring Flow Settings for the new user interface
- Creating a custom button for the new user interface
- Sorting main categories on the Catalog page
- Location-based Cart
- Customizing Column Settings on the Cart and Installed Products (Assets) page
- Customizing number of products displayed on the Catalog page
- Support for custom attributes Page
- New progress bar on all pages
- Hiding Add to Cart and Configure buttons for Option Products
- Displaying all the leaf nodes on the Catalog page
- Cloning of sub-bundle, options and its attributes on the Options page

As compared to the existing setup, the following features are not introduced in this release:

Feature	Description
Dependent Picklist for Attributes	Salesforce Dependent Picklist is not yet supported on the new CPQ user interface. Instead of using SalesForce Dependent Picklist, you can drive the product attribute selection using the Attribute Value Matrices of Apttus CPQ. By using Attribute Value Matrices, you no longer need to maintain the attribute relationships in the SalesForce object and instead you can use admin interface.
	Price breakdown information icon is not displayed for the base price on the Cart page.

Feature	Description
Price Breakdown for cumulative tiers and related pricing	
Custom Labels support for Custom Display Actions	New UI does not support the creation of a new label. You need to use one of the available Custom Action labels and change the key in the Display Action Setting. You should change the Custom Display action name under the Display Action Setting and refer to one of the standard “Custom Action” labels that are provided by the managed package.

The new user interface allows you to select products, configure their attributes and options, define pricing and make adjustments, select installed products, Change, Swap and Cancel orders on one single, user-friendly page.

Configuring Flow Settings for the new user interface

In the AngularJS user interface, flow settings are required so that appropriate pages are loaded for various sections (such as Catalog, Product Configuration, Cart, and Asset) are loaded. In the AngularJS user interface, you can configure options and attributes in a single page.

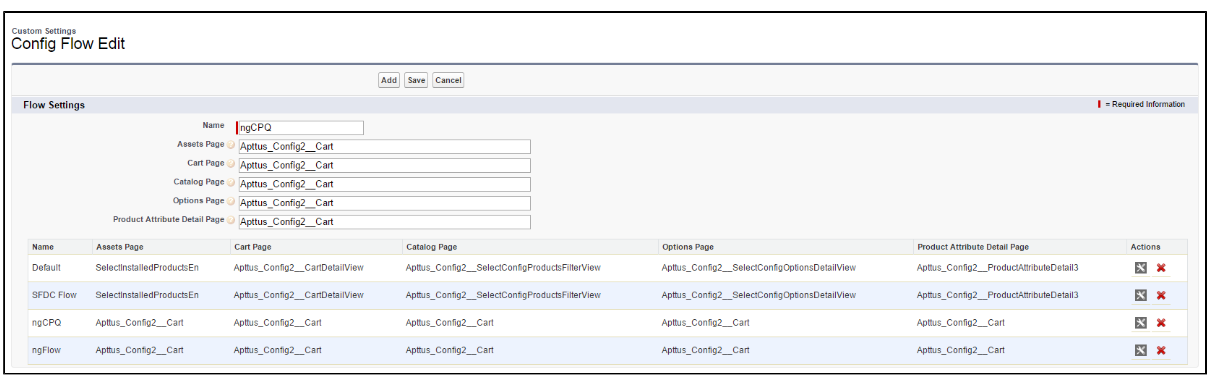
To configure Flow Settings for the new user interface

1. Go to **All Tabs ( ) > Config Settings** and click **Flow Settings**. The page to create a new Flow setting opens.

2. Create a new record for the flow setting.
   - Or -

   Edit an existing record by clicking  in the **Actions** column.
3. Enter requisite information as explained in the following table:

Field	Description
Name	Enter a mandatory name for the flow setting. For example, enter ngFlow since we are defining the flow for the new user interface.
Assets Page	Enter the name of the VF page for displaying the Installed products or Assets. For example, *Apttus_Config2__Cart*.
Cart Page	Enter *Apttus_Config2__Cart* Visualforce page to access the new Cart page.
Catalog Page	Enter *Apttus_Config2__Cart* Visualforce page to access the new Catalog page.
Options Page	Enter *Apttus_Config2__Cart* Visualforce page to access the new Options page.
Product Attribute Detail Page	Enter *Apttus_Config2__Cart* Visualforce page to access the new Product Attributes page.



4. Click **Add** if you have created a new record for flow setting.
   - Or -
   Click **Save** to save your modifications to existing flow setting record.

Once saved, you can also configure the columns and buttons for the different pages in your custom flow by configuring the **Display Column Settings** and **Display Action Settings**.

Creating a custom button for the new user interface

To use the new AngularJS user interface for CPQ as per the flow settings, you must create a custom button for the Quote/Proposal object.

1. Go to **Setup > Create > Objects** and search for **Quote/Proposal** object.
2. In the **Custom Fields & Relationships** related list, click **New**.
3. From **Step 1. Choose the field type**, choose **Formula** as Data Type and click **Next**.
4. From **Step 2. Choose the output type**, Type *Configure (Flow)* in the **Field Label**. Choose **Text** as **Formula Return Type**.
5. In **Step 3. Enter formula**, paste the following URL under the **Simple Formula** tab and click **Next**:

```
IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 , HYPERLINK("
/apex/Apttus_QPConfig__ProposalConfiguration?id="&Id&"&flow=ngFl
ow" , IMAGE("/resource/Apttus_QPConfig__Button_Configure", "Conf
igure Products"),"_self"), NULL)
```

> Ensure that you enter the name of your Flow setting in the *flow* parameter in the above formula. In our example, it is *ngCPQ*.

6. Select appropriate Field Level Security and click **Next.**
7. Choose the page layouts on which you want to display this field.
8. Click **Save**.

You can use productOrderByClause parameter to sequence the products within a Category. Refer to Sequencing Products within a Category for more details.

Accessing the cart in read-only mode

Apttus CPQ enables you to create a Cart in which your Sales rep can just view the product configurations and not make any changes to the field values for the Product Line Items. In order to achieve this functionality, you must specify a new parameter *mode=readOnly* while creating a custom button to launch the New UI.

```
IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 ,HYPERLINK("/apex
/Apttus_QPConfig__ProposalConfiguration?id=" &Id& "&flow=ngFlow"&"&mo
de=readOnly",IMAGE("/resource/Apttus_QPConfig__Button_Configure", "Co
nfigure Products"),"_self"), NULL)
```

By applying this formula to your Configure button, the selection boxes of products are grayed out in the Cart. This prevents your sales rep from applying any mass updates to the products in the Cart.

Sorting main categories on the Catalog page

Using this feature, you can decide the sequence of the main categories. You can control the sequence of the category using the new field called **Sequence** on the Price List Category record under the Category. This helps you to customize and have a better control over the way your categories are sequenced on the Catalog page.

You must have categories created for grouping the bundle and standalone products.

To implement this enhancement:

1. Go to **All Tabs** ( ), search for **Categories** and click on it.
2. Select the main category you want to sequence.
3. In your category record, look for **Price List** related list.

4. Click **Edit** for an existing price list category.
   - Or -
   Click **New Price List Category** to create a new record for associating your category to the price list.
5. In the **Sequence** field, enter the number in which you want to sequence the selected category.
   If you are creating a new price list category record, select the required price list.
6. Click **Save**.
7. Perform the above steps for all the necessary main categories.

All the main categories will appear on the Catalog page in the sequence you defined.

Location-based Cart

Just like Enterprise CPQ, the new AngularJS based CPQ also allows grouping of products based on the location of the customer. These locations are in sync with the locations specified at the Opportunity or Account from where you create a Quote. Sales representatives can easily differentiate between products according to the location of the customer. The filtration of products is performed before loading the Cart page.

Once you save or finalize your configuration in the Enterprise CPQ, when you configure the products in the new AngularJS based cart, you have the option to group the products by location or product.

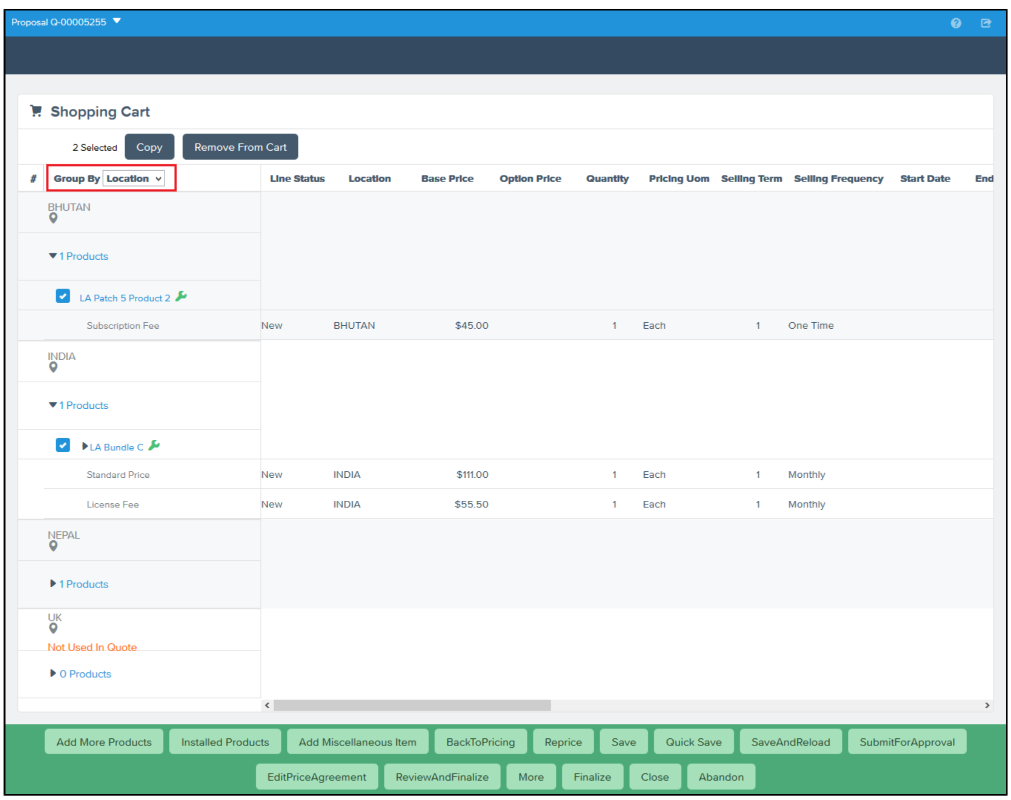
> **Note**
>
> Ensure that you have selected **Enable Location** check box available at **Setup > Develop > Custom Settings > Config System Properties > Manage > System Properties**.

To enable this feature in the new AngularJS based cart,

1. On your Quote/Proposal record page, search for **Proposal Location** related list and click **New Proposal Location**.

2. Create records using the look-up icon, in sync with the locations specified in the Account or Opportunity from which you have created your Quote/Proposal. For example, USA, India, UK, and China.
3. Click **Configure Products** for your Enterprise CPQ.
4. On the top-right corner, use the **Location** drop-down list to configure the products for different locations and add them to the Cart. For example, choose 2 products for the India location, 3 products for the US and so on.
5. After configuring the options and attributes for the chosen products, click **Save** on the Cart page.
6. Once you return to the Quote/Proposal detail page, click **Configure Products (new-CPQ)** or the button which you have defined for configuring the products using the new AngularJS CPQ.
7. Just above the products list, choose **Group By > Location** to see the products filtered according to the configuration you did in Step 4. You will see the products differentiated under each location.

   Not Used in Quote under a location indicates that either no product is linked to the Quote /Proposal or you have not created a **Proposal Location** record for this.



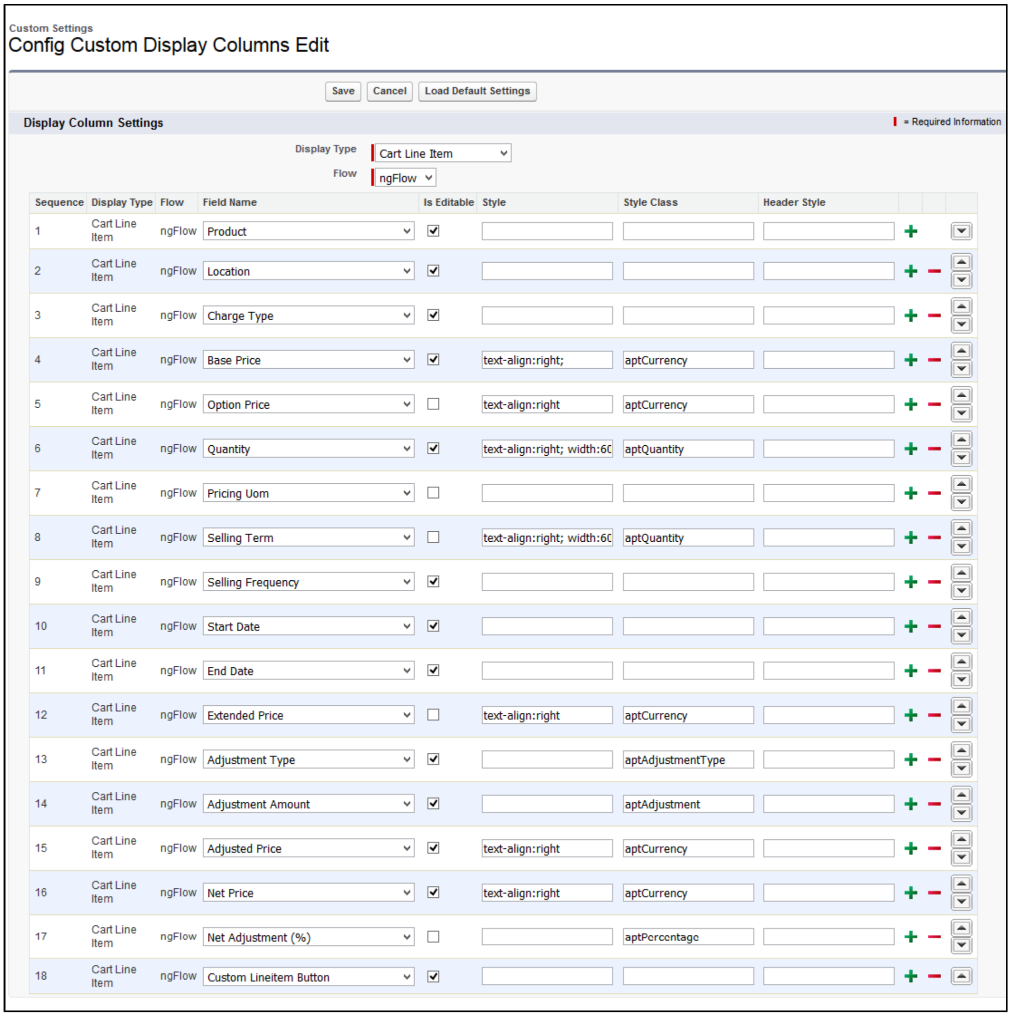
> **Important**
>
> Ensure that you have created the **Proposal Location** records in sync with the locations of your Account or Opportunity, else you may not see the product grouping by location.

Customizing Column Settings on the Cart and Installed Products (Assets) page

The custom settings for columns defined at **All Tabs > Config Settings > Display Column Settings** are now applicable for the new AngularJS CPQ as well. Using this custom setting, you can control column style, column header, sections such as Cart Line Items and Totals which are displayed on the Cart and Installed Products (Asset) page.

To use the above custom settings, you must have defined a flow for the new AngularJS CPQ under **All Tabs > Config Settings > Flow Settings**.

1. Go to **All Tabs > Config Settings** and click **Display Column Settings.**
2. From the **Display Type** drop-down list, select the required type in which you want to customize the columns. For example, if you want to customize the columns in the main Cart Line Item section, containing columns such as Product Name, Charge Type, and more are shown, choose *Cart Line Item*.



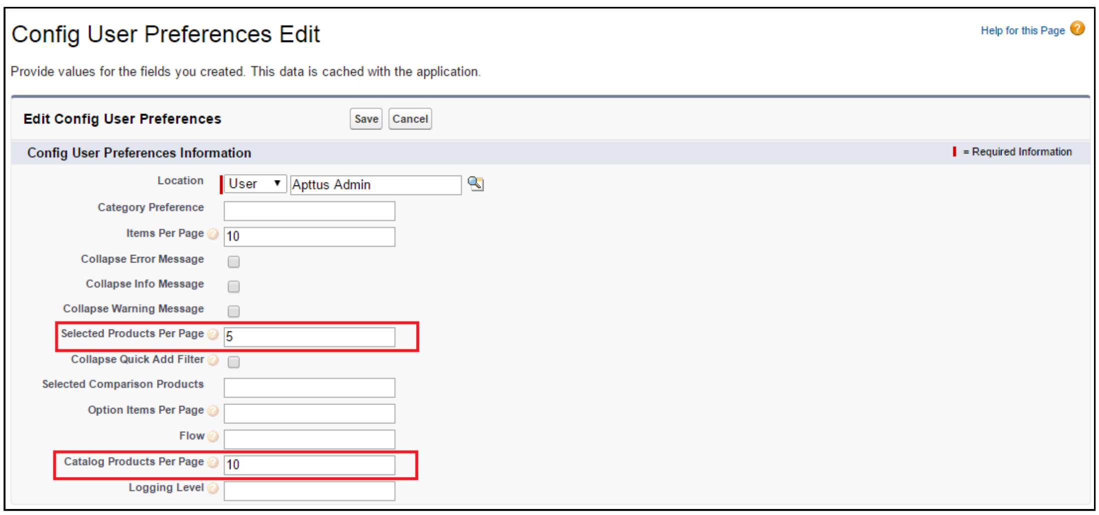
3. From the **Flow** drop-down list, choose the name of the Flow settings you have defined under **All Tabs > Config Settings > Flow Settings**.
4. Specify the requisite information in the **Style**, **Style Class** and **Header Style** for each field, as per your requirement.
5. Click **Save**.

Customizing number of products displayed on the Catalog page

For the New UI, customize the following custom setting to display the number of products on the Catalog page for a specific user. The products are divided across the pages using the Pagination concept.

To customize the user preferences

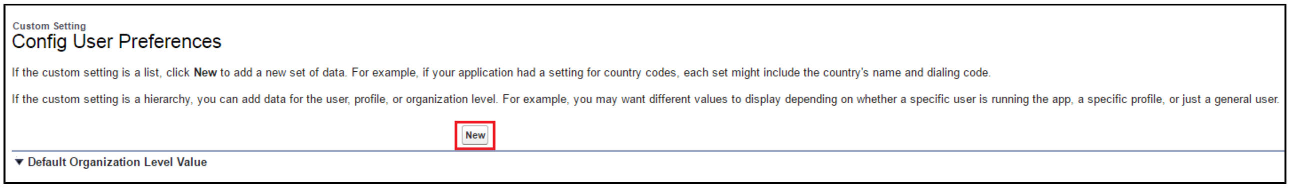
1. Go to **Setup > Develop > Custom Settings** and search for **Config User Preferences.**
2. Click **Manage** next to **Config User Preferences**.
3. For the required user, do any of the following:
   Click **View** to view the details and click **Edit**.
   - Or -
   Click **Edit** to edit and make changes to the existing user preferences.



4. In the **Catalog Products Per Page** field, enter the number of products that you want to display on the Catalog page per page. Valid values are *10*, *20*, *50*, and *100*. This setting is available, especially for the new user interface.
   You may also customize the number of products in the mini cart on the Catalog page by entering the requisite number in the **Selected Products Per Page** field. Valid values here are *5*, *10*, *15*, *20*, and *25*. This setting is available for both, the enterprise CPQ and the new user interface in CPQ.
5. Click **Save**.

When you browse the catalog page as a specific user, you will find the specified number of products on a single page and the rest of the products are divided into multiple pages using Pagination.

If you want to set up the above settings at an org level, click **New** on the Config User Preferences page.



Support for custom attributes Page

You can now use your pre-defined custom attribute page while navigating from the Catalog page, Mini Cart, Installed Products (Assets) page, and the Cart page in the AngularJS based CPQ UI framework . By default, the system chooses the *Apttus_Config2__ProductAttributeDetail3* VFpage. This capability allows the customers to have their own UI implementation for the product configuration experience using underlying Apttus APIs. This enhancement is useful when you have designed a custom attribute page with the new look and design and want to use it while configuring the attributes.

To use this enhancement, customize your flow settings under the Custom Settings as described below.

1. Go to **All Tabs ( ) > Config Settings** and click **Flow Settings**.
2. In the **Product Attribute Detail Page** field, enter the name of your custom attribute page.
3. Click **Save**.

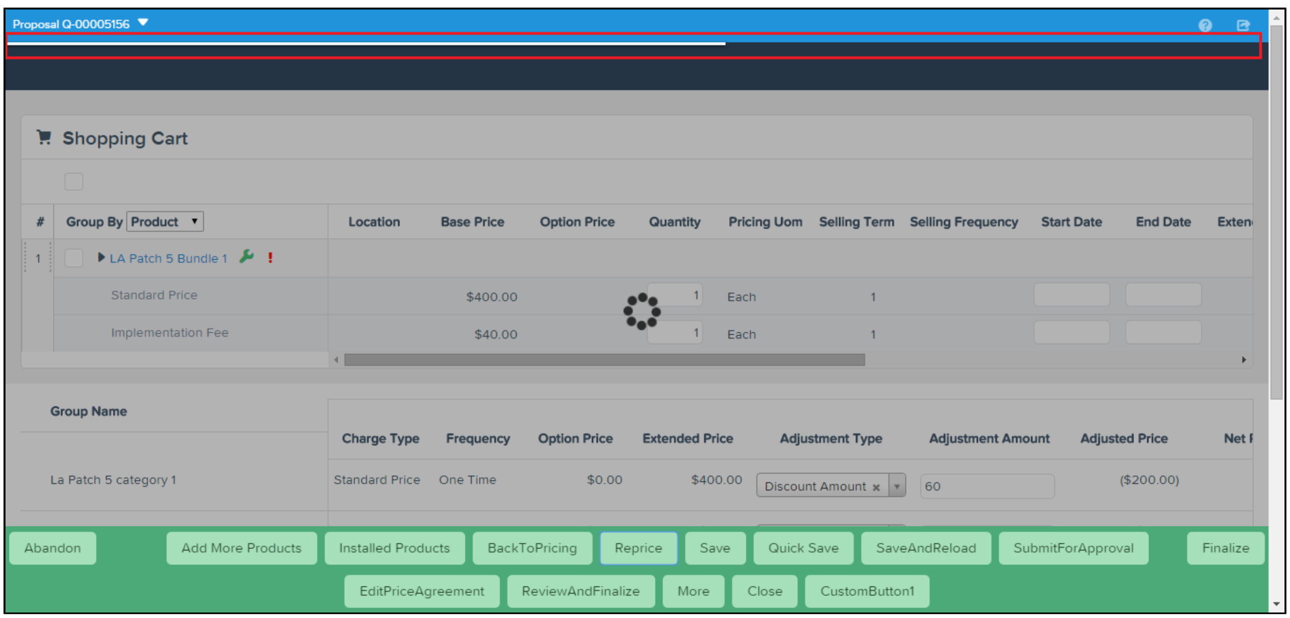
Now, when you configure your attributes page by clicking **Configure** from the following pages, you will see your customized attributes page.

- **Configure** button on the Catalog page.
- Wrench icon on the Mini Cart, next to the respective product on the Catalog page.
- **Configure** button on the Installed products (Assets) page.

- Wrench icon for configuration under the Actions column, next to the respective product on the Cart page.

New progress bar on all pages

A thin, white colored progress bar is displayed, just below the main header, whenever you click **Configure** or **Go To Pricing** on any of the Catalog, Attributes, Options or Cart page. This indicates that the system is performing timed actions based on your clicks. Completion of the progress bar indicates that the system has completed the asynchronous tasks.



This progress bar is visible when you click **Configure** from the following pages:

- Configure button on the Catalog page.
- Wrench icon on the Mini Cart, next to the respective product on the Catalog page.

- Wrench icon for configuration under the Actions column, next to the respective product on the Cart page.

Hiding Add to Cart and Configure buttons for Option Products

In order to prevent your sales representative from mistakenly selling the option products without a bundle, you can now hide the **Add to Cart** and **Configure** buttons from the Catalog and Bundle pages. This enhancement is also useful when your customer searches for an option product using the Global search and tries to add it to the cart without buying its associated bundle product. This creates a clear difference between the standalone and option product on the Catalog page.

With this feature, for each option product, the text **Sold as Option** is displayed when you add the option product from the Catalog page without its bundle product.

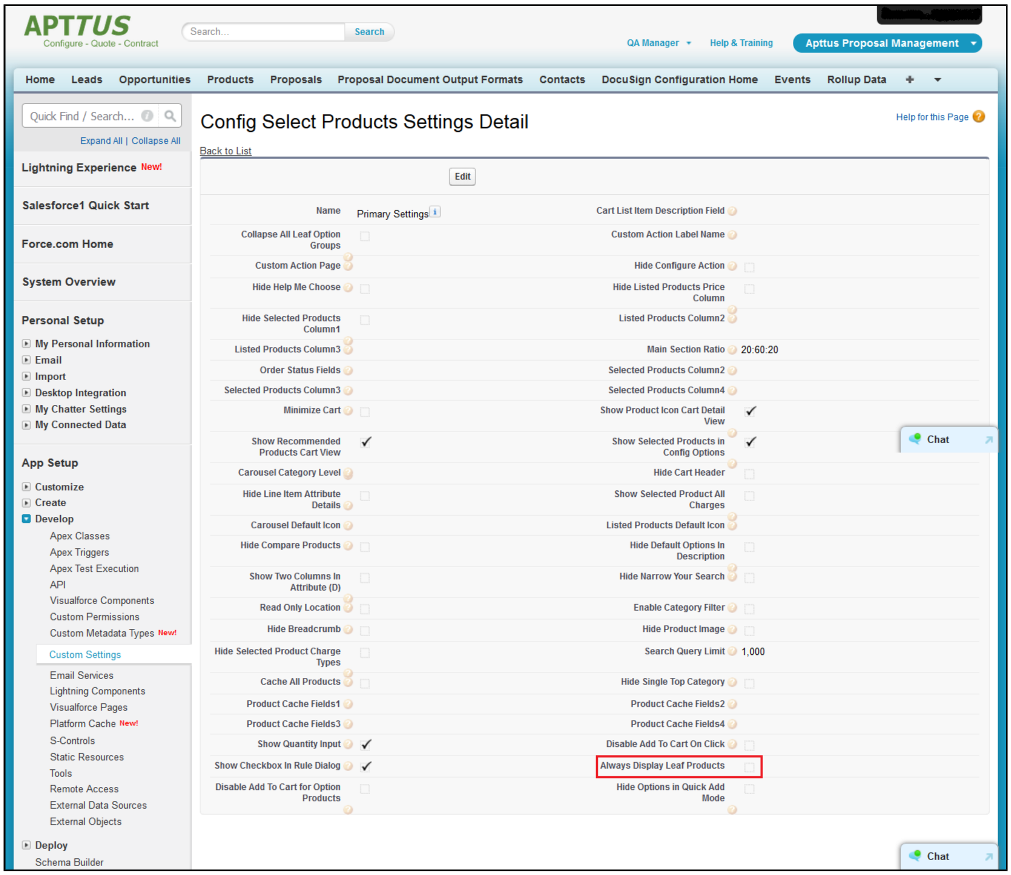
To use the above enhancement, go to **Config Settings > Catalog Page Settings** and select **Disable Add To Cart for Option Products**. For more details, refer to Catalog Page Settings.

Displaying all the leaf products on the Catalog page

The new UI supports the custom setting **Always Display Leaf Products** to load and display all the leaf nodes for a specific category. These leaf nodes have products associated with them. With this feature, it is easier for a sales representative to just click the main category or any sub category to load all the sub-categories (and products) at one click. This feature is already available in the Enterprise CPQ.

To configure this setting,

1. Go to **Setup > Develop > Custom Settings > Config Select Product Settings**.
2. Next to **Config Select Product Settings,** click **Manage.**



3. Select the **Always Display Leaf Products** check box.

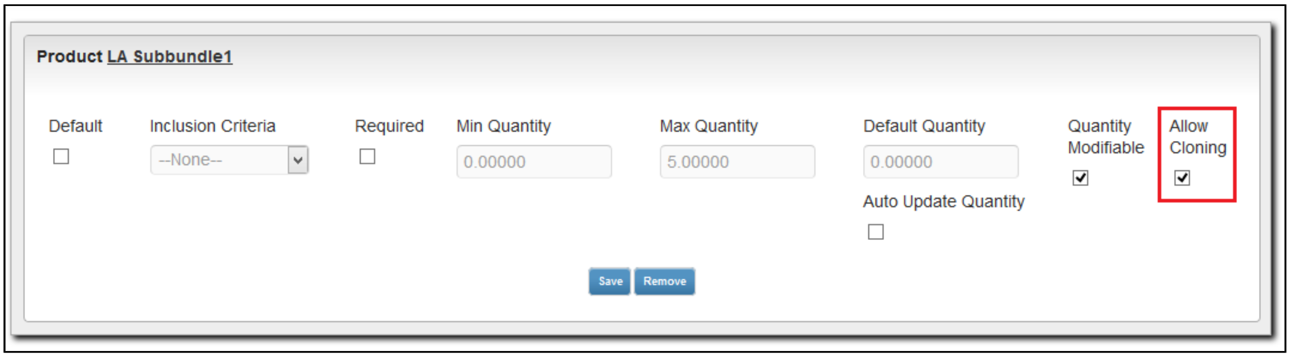
Now, if you click any category on the Catalog page, all its leaf products are displayed by default.

Cloning of sub-bundle, options and its attributes on the Options page

This feature allows to copy the options and attributes of a bundle product.

If **Allow Cloning** check box is selected for a specific product option or sub-bundle, the system will show clone icon next to the option on the configuration page while configuring the product. When you click this icon, system will clone all associated attributes as is.

This is helpful when you want to replicate the same options and its attributes under a single sub-bundle product. A single icon click makes it easier for the sales representative to clone all of these at once.



Multiple Adjustments at the Line Item

Sales reps can apply multiple adjustments at the line level either by applying the adjustments in the corresponding columns or clicking the **$** icon next to the line item from the cart page.

You can allow the sales rep to create more than one adjustment only if the Max Adjustment Lines custom setting has a value greater than 1. This setting is available at **All Tabs > Config Settings > System Properties > Pricing Settings > Max Adjustment Lines.**

The following table describes the different values of this setting and its behavior.

Max Adjustment Lines Value	Behaviour
Blank field	$ sign does not appear.
1	$ sign appears against the line item. However, you need to use the adjustment type and adjustment amount columns in the line item itself.
>1	$ sign appears against the line item. You can click on the $ sign to open a multiple adjustment popup. You can create as many adjustments as the value defined in the Max Adjustment Lines.

To customize the fields displayed in the multiple adjustment popup, navigate to **Config Settings > Display Column Settings**. Select *Adjustment Line Item* from the **Display Type**, and *ngFlow* from the **Flow** drop down.

You can also display custom fields that you have defined for Adjustment Line Item object on the cart by using the above procedure.

Enabling Price Breakup

The price breakup enables the sales rep to analyse the price breakup for any product on the Cart. A small icon () next to Base Price is available. Clicking this icon shows the price breakup as defined in the Price Matrix.

To enable Price Breakup for products

To create a Price Matrix, you must have already created a Price Dimension.

1. Go to your Product and click **Product Console.**
2. Click on the PLI record and click **Edit**.
3. Under the **Matrices** tab, click **New.**
4. Select **Matrix Type** as *Dimension* and select your Price Dimension in the **Dimension 1** drop-down list.
5. In the **Dimension 1 Value Type** drop-down, select *Cumulative Range*.
6. Click **Close** to close the Matrices.
7. Click **Save** to save the PLI.

Displaying attributes and options as inline or on a new page

For the new products (bundle or sub-bundle), you can choose to display the options and attributes as inline or on a new Visualforce page. A new drop-down list, **Config Type** is introduced to indicate this enhancement for options and attributes.

When you associate attributes or options to your product, you can choose to select *Inline* or *New Page*.

- *Inline:* The attributes and options can be configured on the configuration page.
- *New Page:* A wrench icon is displayed near the bundle or sub-bundle product which has to be configured. When you click this icon, a new page is opened for configuration of options and attributes.

By default, *Inline* option is chosen for the new products, for both, options and attributes.

If you do not choose any option from the drop-down list, no attributes and options are loaded for your product.

Note: This enhancement does not affect your existing product configurations and is applicable for the new products only.

Product Management

Products are the primary component of your CPQ system. Using Apttus CPQ, you can create individual products and product bundles. Once you've created products, you can add attributes to the products. Attributes allow you to specify product options, such as color, size, or model.

Using feature sets, you can also configure Apttus CPQ to allow your users to compare the features of multiple products. By implementing feature sets, you allow your users to compare products and choose the product that best meets their requirements.

Working with Products

Products are standard products or services you sell to your customers and are defined in the standard Salesforce products table. A product is classified as one of the following types: Standalone, Bundle, or Options. This section shows you how to create products and then group the products to make them easier for customers to find.

A standalone product can be sold on its own under the offering category; whereas options can be grouped together as an option group and then associated with a bundle product to be categorized as an offering. You can also associate a bundle with another bundle.

The following sections describe various product operations, starting from creating the product to associating the product.

- Creating Products—In Apttus terminology, a product is a product or service that can be sold on its own as a standalone product or sold as an option for other products. Products or services are set up as standalone products, bundled products, or options for other products.
- Associating Option Groups to a Bundle—You must associate an Option Group that contains Options to a Bundle. After associating Options to an Option Group, you must associate the option group to a Bundle.
- Associating a Product to a Category—A product (standalone or bundle) must be associated to a category to be displayed on the catalog page.
- Configuring Multi-level Bundles—Your business may require you to configure a bundle in a bundle. This offers increased control over product selection, as well as making it easier to select products because selecting one bundle full of products can replace having to select those products separately.

- Configuring Numeric Expressions—Numeric expression is a replacement for Salesforce Formula functionality and it is evaluated run-time at the time of rule execution.
- Running a Bundle Maintenance Batch—After making changes to any bundle product association, you must run a bundle maintenance batch that synchronizes all the bundles and options.
- Using a Product Family—Product Family is a standard salesforce.com feature. You can use the Product Family picklist to categorize your products. For example, if your company sells both hardware and software, you can create two product families: Hardware and Software.
- Configuring a Product Group—Product Group (as opposed to Product Family) can consist of products from different Product Family.
- Cloning a Bundle within the same Quote—You can clone a bundle within the same quote, make modifications, and use it as another bundle.
- Expanding the Batch Jobs Section—The Batch Jobs section can be expanded to display all the batch jobs run.

Creating Products

A Product in Apttus terminology is a product or service that can be set up to be sold on its own as a standalone product or options of other products. Products or services are set up as standalone, bundle, or options of other products.

- Standalone - This is an individual product that can be sold on its own.
- Bundle - Any product that has other products (options or bundles) associated with it.
- Options - This is a product that can be sold with a bundle product only via an option group.

To create a product

1. Navigate to the **CPQ Console** tab.
2. From the Product Management section, click **Add Products**.
3. Type a mandatory **Product Name** and type a **Product Code**. The Product Code can be an internal code or a product number to identify the product within your organization.
4. Type a **Product Description** and select a **Product Family**.

Note

If the required Product Family is not displayed in the list, you must add a picklist value in the Product Family field on the Product object.

5. To activate the product, select **Active**.
6. Enter the following product information, as required.

Option	Description
Must Configure	Indicates that the end user has to configure a product to add options or attributes or both. The Configure button is the only option available on the Catalog page. When you select Must Configure, you must also select either **Has Options**, **Has Attributes**, or both. Clearing the check box allows an end user both the options; either add the product to cart directly or configure the product with attributes or options, or both.
Has Attributes	Indicates that the product has attributes associated to it. During the quote creation process, the end user can enter values for those attributes. Using the product attributes administration tool, you can associate product attributes to products.
Has Options	Indicates a product as a bundle that has options (bundles/options). Using the bundle manager tool, option products can be associated with a bundle product.
Has Defaults	Indicates that when a product is added to the shopping cart, there are some other products that may get automatically added to the cart.
Configuration Type	Select one of the following type of the product: • **Standalone** • **Bundle** • **Option** This field is mandatory.
Uom	Select a unit of measurement.

Option	Description
Has Search Attributes	Indicates that the product has search attributes and is useful for searching related products based on product attribute values.

7. Click **Save**.

A product is created and saved.

> **Note**
>
> You can associate a product to a category or a price list. If you created an option product, you must first associate it to an option group.

Associating Options to Bundles

An Option is associated with a bundle product only through an Option Group to give the end user an additional choice for a product. To create an Option product, create a new product and from configuration type select Option. You must associate an Option Group that contains Options to a Bundle. After associating Options to an Option Group, you must associate the option group to a Bundle.

> You must have an existing Option product and an Option Group category.

To associate Options to Option Groups

1. Navigate to the CPQ Console tab and from the Product Management section, click **Manage Categories**.
2. Select an Option Group and click **Hierarchy Manager**. For more information on how to build a hierarchy, see Working with Hierarchy Manager.
3. Select the lowest level node and select an option product from the available products list or search for an option product to associate with the option group.
4. Click **Associate**. You can associate multiple products to a category.

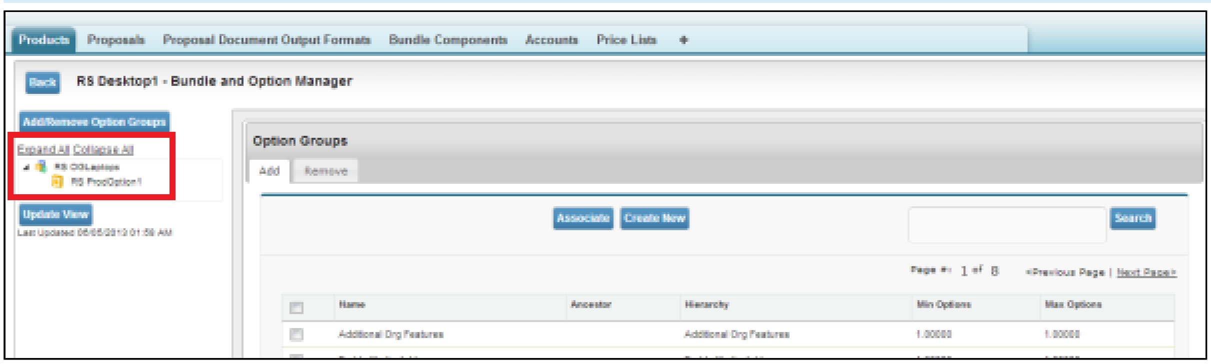
5. After you have selected the products to be associated, click **Save**. The selected products are listed in the **Remove** tab of the product classification section.

Option products are associated with an option group. Next, you must associate option group to a bundle product through an option group.

To associate Option Groups to a Bundle

1. Navigate to the CPQ Console tab and click **Manage Products**.
2. Select the bundle product to which you want to associate option groups and click **Product Console**.
3. From the **Catalog** section, click **Manage Bundles/Options**.
4. Click **Add/Remove Option Groups**.
5. From the available option groups list, select or search an option group to associate to the product and click **Associate**.
6. To update the associated option group list in the left pane, click **Update View**.

> Whenever you edit the Bundle or Option from the Bundle or Option Manager, clicking Update View, updates the bundles and options in the fast mode within the same page.



7. Click **Back** to return to the Product Console.

Options are now associated to the bundled product through an option group. You can associate the bundle product to a category or a price list.

You can choose to lock the option quantity value, which is populated using numeric expressions.

To modify an existing Option Group within a Product

1. Navigate to the CPQ Console tab and click **Manage Products**.
2. Select the bundle product to which you want to associate option groups and click **Product Console**.
3. From the **Catalog** section, click **Manage Bundles/Options**.
4. Click an existing option group which you want to modify.
   Fill in the required details as explained in the table below:

Field Name	Description
Content Type	Options - Default value for Content Type is Options. Attributes - When content type is selected as Attributes, attribute group lookup filed is displayed. Detail page - When content type is selected as Detail page, the system will display the text field in which you can specify the detail custom page URL. Note: When you migrate data from one org to another using Data loader, ensure that you select Options value for the Content Type.
Min Options	Specify the minimum product options which a user should select on the options page.
Max Options	Specify the maximum product options which a user should select on the options page.
Min Total Quantity	Specify the minimum required value, inclusive of all the options inside an option group. For example, if you enter 10 in this field, the minimum quantity of all the options should reach 10. Thus, you can have 5 options, each with quantity=2.

	You can also use Numeric Expressions to populate this field. Ensure that the return value of your numeric expression is of type *Number*.
Max Total Quantity	Specify the maximum required value, inclusive of all the options inside an option group. For example, if you enter 10 in this field, the maximum quantity of all the options should reach 10. Thus, you can have 5 options, each with quantity=2. You can also use Numeric Expressions to populate this field. Ensure that the return value of your numeric expression is of type *Number*.
Is Hidden	Select this check box if you want to hide this option group on the catalog page.
Is Picklist	Select this check box when you want to display your options in the form of a picklist, from which your user can select only one option. When you select this check box, ensure that Min Options and Max Options are set to 1.
Modifiable Type	Variable – Select this option if you want to allow the user to change the quantity of the option products during the configuration. Fixed – Select this option if you want to restrict the user from changing quantity of the option products inside this option group.

Note that you should cautiously use the Min/Max option field in conjunction with hiding an option group . If no option products are visible due to Visibility rules configuration or different price lists, and you have set the minimum options as zero, the system considers this configuration as invalid.

5. Click **Save**.
6. To update the associated option group list in the left pane, click **Update View**.

To modify an existing Option Product inside a Bundle

1. Navigate to the CPQ Console tab and click **Manage Products**.

2. Select the bundle product to which you want to associate option groups and click **Product Console**.
3. From the **Catalog** section, click **Manage Bundles/Options**.
4. Click an existing option to modify.
   Fill in the required details as explained in the table below:

Field Name	Description
Default	Select the check box to mark this option as Default.
Inclusion Criteria	Specify the minimum product options that a user should select on the catalog page.
Required	Select the check box to mark this option as mandatory.
Min Quantity	Specify the minimum quantity for the specific option. You can also use Numeric Expressions to populate this field.
Max Quantity	Specify the maximum quantity for the specific option. You can also use Numeric Expressions to populate this field.
Default Quantity	Select the check box if you want to add a specific number of options to the bundle during configuration. You can also use Numeric Expressions to populate this field.
Quantity Modifiable	Select this check box to make the option quantity modifiable by the user on the Options and Cart page.
Allow Cloning	Selecting this check box for specific product option, bundle or sub-bundle will show the clone icon next to the option on the configuration page while configuring the product. When you click this icon, the system will clone all associated attributes as is.
Auto Update Quantity	

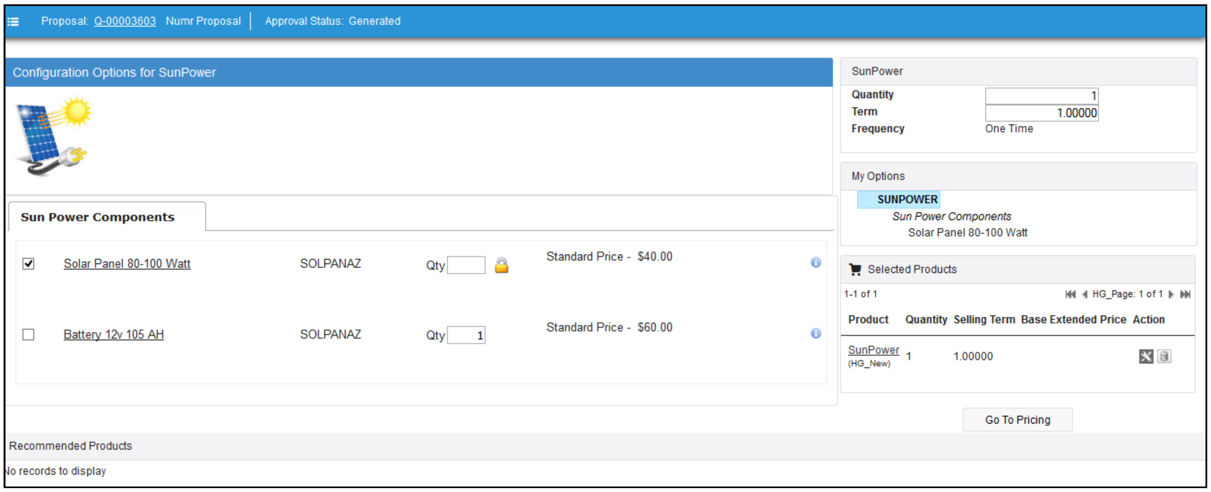
	Select this check box to lock an option quantity when you have the option quantity being derived from a numeric expression. Selecting this check box will show the lock icon next to the Quantity filed on the Catalog page to indicate you cannot modify the Quantity because it is derived through a numeric expression.
Config Type	Choose if you want to display the options as inline or on a new Visualforce page. This is to ease the sales rep to configure the options on the same page or on a different configuration page. By default, *Inline* option is chosen for the new products, for both, options and attributes. *Inline:* The attributes and options can be configured on the configuration page. *New Page:* A wrench icon is displayed near the bundle or sub-bundle product which has to be configured. When you click this icon, a new page is opened for configuration of options and attributes.

5. Click **Save**.
6. To update the associated option group list in the left pane, click **Update View**.

To lock an Option value

Ensure that the quantity field derives its quantity from an expression built using the field expression builder.

1. To lock an option quantity, navigate to the **Option Group**, choose your option and select **Auto-Update Quantity**.
2. On selecting the check box, a lock icon appears next to Quantity, when you configure your Bundle from the **Catalog Page**. The user can click the lock icon next to an option, to ensure that any updates to the field from which its value is derived does not affect the value set the first time. You can lock the quantity field for an option product, whose quantity is populated based on multiple attribute values using the field expression builder. Once the user enters the attribute value on the **AttributeDetail3** page and clicks Next, the value for quantity is auto-populated.
3. Select the option and click the **Lock** icon.



4. Navigate back to the Attribute detail page and edit the attribute value, then click **Next**. Notice that the value of the Option does not change even though the attribute value has changed.

Associating a Product to a Category

A product (standalone or bundle) must be associated to a category to be displayed on the catalog page.

This can be achieved from the product console. The product console allows you to associate a product to a category through a user-friendly interface. The product console helps you to administer product properties and associate products with other CPQ artifacts. You can also associate a product to a category from the Hierarchy Manager.

To associate a product to a category

1. Select an existing product and click **Product Console**.
2. From the catalog section, click **Associate to Product Category**. All the available categories are displayed along with their hierarchy.
3. Select the category to associate your product with and click **Associate**. The selected category is displayed in the Associated section.
4. Click **Done**.

The product is now associated with a category.

For performance implications, Apttus recommends that you limit the number of products in a single category to 1000.

Product Visibility

When you go to configure products for a Quote/Proposal, you can control which products from the price list are visible on the catalog page.

The following table describes some scenarios through which you can control the product visibility on the catalog page.

Scenario	Functionality	Configuration Required
You want ensure that the products having Red color are visible on Catalog page.	Refine Your Search	Select **Red** check box in the color section in Refine your Search dialog
	Search Filters (CPQ)	Select the appropriate price list in the Inclusion Criteria; select *Color* fr *Equal to* from Operator, and *Red* from Value.

	Visibility Rules through Custom Classes	Modify the custom callback class code to list the catalog products as
		`global String getProductFilterExpr(CustomClass.ActionPara`
		`        return 'Color__c = \'Red\'';`

You want to ensure that the Bundle	Search Filters (CPQ) and Visibility Rules	Select the appropriate *Price List* in the Inclusion Criteria; select *Config* from Field, *Equal to* from Operator, and *Bundle* as Value. Modify the custom callback class code to list the catalog products as

Scenario	Functionality	Configuration Required

products having Red color and relevant Option products having Green color are visible on Catalog page.	through Custom Classes	`global String getProductFilterExpr(CustomClass.ActionPara`
		`return 'Color__c = \'Red\'';`
		
		Modify the custom callback class code to list the option products as
		
		`global String getOptionFilterExpr(CustomClass.ActionPara`
		`return 'ComponentProductId__r.Color__c =\'Green\'`

Product Visibility can be controlled in the following ways.

- Using Refine your Search functionality
- Using Search Filter (CPQ) functionality
- Using Visibility Rules through Custom classes functionality

Search Filters (CPQ)

When you configure products for a Quote/Proposal, you can control which products from the price list are visible on the catalog page based on search filters, where you set up the criteria of your choice.

Within a search filter, you can select the header level fields of object records as Entry Criteria for deciding whether a price list's products will be filtered to a certain subset of the full list. This subset also uses the header level fields of object records to determine which products are displayed.

The system checks whether a search filter is applicable to the price list you are using whenever you go to the product selection page. When you click **Configure Products** in the Quote/Proposal, the search filter is executed.

If you do not want entry criteria to be determined by Quote/Proposal header level fields, it can be determined by using the user record.

To configure Search Filters (CPQ)

1. Go to **Setup > App Setup > Create > Objects** and select **Search Filter (CPQ)**.
2. From the Custom Fields & Relationships section, select **Business Object**.
3. From Picklist Values, click **Del** to remove SearchAttributeValue__c. This value must be deleted, if it has not already been deleted. Do not delete *Apttus_Config2__SearchAttributeValue__c*, because it is the correct search value to use.
4. From Picklist Values, click **New** and enter the API names for the objects you want to be able to filter on. This could include:
   - *Asset*
   - *Apttus_Config2__AssetLineItem__c*
   - *Product2*
5. Go to **Setup > App Setup > Create > Objects** and select **Search Filter (CPQ)**.
6. From the Custom Fields & Relationships section, select **Filter Type**.
7. From Picklist Values, click **New** and enter the names you want to use to group the Business Object picklist values. If you wanted to have names to match the values entered in step 4, you could include:
   - *Asset*
   - *Product*
8. From Field Dependencies, click **New**.
9. For Controlling Field select Filter Type and for Dependent Field select Business Object. Click **Continue**.
10. Select which business object fields should be associated with which filter and click **Save**.

Showing Columns: 1 - 3 (of 3) < Previous | Next > View All Go to

Filter Type:	Product	Asset
Business Object:	Apttus_Config2__SearchAttributeValue__c	Apttus_Config2__SearchAttributeValue__c
	Asset	**Asset**
	Apttus_Config2__AssetLineItem__c	**Apttus_Config2__AssetLineItem__c**
	Product2	Product2
	Apttus_Config2__ProductConfiguration__c	Apttus_Config2__ProductConfiguration__c
	Apttus_Config2__ProductAttributeValue__c	Apttus_Config2__ProductAttributeValue__c

Showing Columns: 1 - 3 (of 3) < Previous | Next > View All

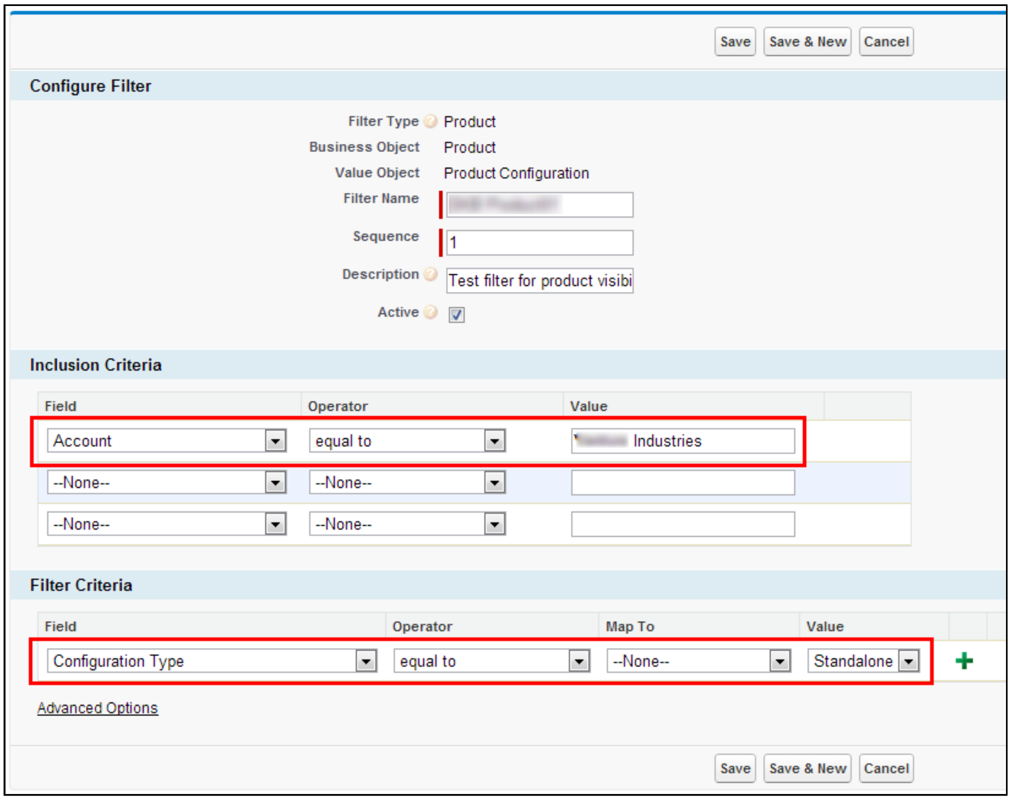
The settings have now been configured so that you can create Search Filters (CPQ).

Click ✚ and select **Search Filters (CPQ)** to create the filter records, required to limit product visibility.

To create a Search Filter (CPQ)

1. Click ✚ and select **Search Filters (CPQ)**.
2. Click **New** to display the New Search Filter page.
3. Select a **Filter Type** to limit which objects will be available in the *Business Object* list.
4. From Business Object, select which object you want to use to control the fields available for **Filter Criteria**.
5. From Value Object, select which object you want to use to control the fields available for **Inclusion Criteria** and click **Next**.
6. Enter a **Filter Name**, **Sequence** (which is used to determine the order in which the search filters are evaluated), and **Description.**
7. In the Inclusion Criteria section, select the fields and expressions you want to use as entry criteria for the filter. You can use a maximum of three fields. If you use multiple fields, the relationship between them is an *AND* relationship, meaning each expression must evaluate as true. The available fields are based on the *Value Object*.
8. In the Filter Criteria section, select the fields and expressions you want to use to control which products will be visible when the Inclusion Criteria is met. You can have a standard *AND* relationship between the expressions or use Advanced Options to the fields together in a more complex formula. The available fields are based on the *Business Object*.

> Apttus CPQ does not support the values containing an apostrophe (') in your Filter Criteria.



9. In the above screen shot, the *Inclusion Criteria* requires the *Account* be *Venture Industries* for the Quote/Proposal that the product pricelist is associated with. When this is true, this search filter will be used to control which products are displayed. The products that will be displayed will only be those that have a *Configuration Type* of *Standalone*.
10. Select **Active** check box to activate this search filter.
11. Click **Save**.
12. Run the **Product Filter Criteria Maintenance**.

When someone goes to configure products, the products that are visible to them will be limited to those defined by the Filter Criteria when the Inclusion Criteria is met.

Visibility Rules through Custom Classes

When you go to configure products for a Quote/Proposal, you can control which products from the price list are visible on the catalog page based on custom classes, where you set up the criteria of your choice.

You have to enter the name of the custom callback class that you use in the Custom Classes.

Within a custom class, you can select the entry criteria for deciding whether a price list's products will be filtered to a certain subset of the full list. You can define the criteria to select the standalone products and option products as well.

When you click **Configure Products** from a Quote/Proposal or **Add More Products** from the Shopping Cart, the system checks whether a criteria is applicable for the specific price list.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.

> **Note**
>
> This is just a sample callback custom class to control the visibility of products on the Catalog page.
> The following custom code is written to filter the Catalog products having color Red and the options products having color Green. You may change the custom class to fit your business requirements.

```
 /**
*  Apttus Config & Pricing
*  SampleProductFilterCallback
*
*  @2011-2013 Apttus Inc. All rights reserved.
*/

global with sharing class SampleProductFilterCallback implements
 CustomClass.IProductFilterCallback {
public void
start(ProductConfiguration cart, String productSearchScope,
List<String>
productSearchFields) {

    }

    /**
     * Callback to return part of SOQL filter clause
     * This filter is used in listing catalog products
```



```
     * @param parsms is the CustomClass.ActionParams that
contains accountIds, locationIds when available
     * @return The query filter is like the following.
     * Name LIKE 'A%' AND Quantity__c > 100
     * Id IN ('000123', '000124')
     */
    global String getProductFilterExpr(CustomClass.ActionParams
params) {
       return 'Color__c = \'Red\'';
    }

    /**
     * Callback to return part of SOQLfilter clause
     * This filter is used in listing option products
     * @param parsms is the CustomClass.ActionParams that
contains accountIds, locationIds when available
     * @return The query filter is like the following.
     * Name LIKE 'A%' AND Quantity__c > 100
     * Id IN ('000123', '000124')
     */
    global String getOptionFilterExpr(CustomClass.ActionParams
params) {
        return 'ComponentProductId__r.Color__c =\'Green\'';
    }
}
```

4. Click **Save**.

> **Note**
>
> The Class Name will be auto-populated. If you are using the Apttus Custom Class code then ensure the name is SampleProductFilterCallback.

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** besides **Config Custom Classes**. Click **Edit** for Custom Classes and enter **SampleProductFilterCallback** or the name of your custom callback class in Product Filter Callback Class.

Now, if the user clicks Configure Products, the products that have a Red color in the catalog product and Green color in the Green color will be displayed on the Catalog page.

Category Visibility

Similar to visibility rules for Products, you can apply the visibility rules for the categories displayed on the Catalog page. This enhancement allows you to display only the relevant categories to a Sales rep who sells only a specific set of products from the large catalog.

By hiding only the relevant category, you can still allow your end user to search for the products and add to the Cart.

If you are hiding the product and its parent category, you need to configure two Search Filters, one for the product and another for category.

Each visibility rule type runs in sequence. The rule stops processing after the criteria for the first rule is satisfied. If multiple visibility rules on the product are satisfied, only the first rule gets executed. If one visibility rule on the product and one on the category is satisfied, both rules get executed, since they are of different types.

> For the Sales rep to see the Refine Your Search filters on the catalog page, ensure that you do not hide corresponding leaf categories. Refine Your Search that is applicable for the child category can only be used when such child categories are not made hidden through the Search Filter.

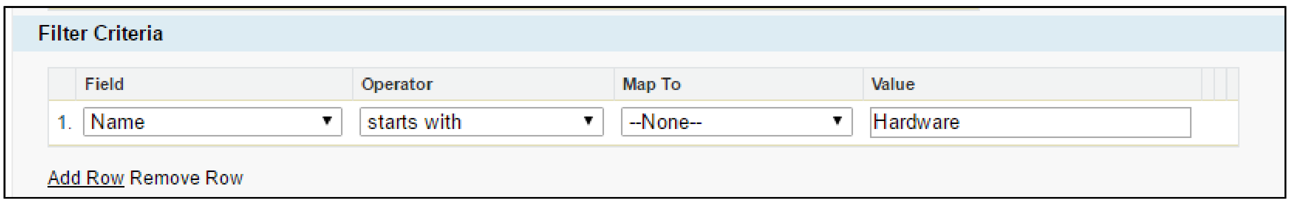
Prerequisites

- In the **Search Filter (CPQ)** custom object, add *Category* as a picklist value for **Filter Type** custom field.
- In the **Search Filter (CPQ)** custom object, add *Apttus_Config2__ClassificationHierarchy__c* as a picklist value for **Business Object** custom field.
- Update the Field Dependency for **Filter Type** picklist to show only *Category Hierarchy* for Business Object when *Category* is selected for Filter Type.

To display the category whose name starts with Hardware

1. Navigate to **All Tabs > Search Filter (CPQ)**.
2. Click **New.**
3. In the Select Object screen, select *Category* as a **Filter Type**, and *Category Hierarchy* as a **Business Object**.

4. Select *Product Configuration* or *User* as a **Value Object** in case if you want to specify Inclusion Criteria in the next screen.
5. Specify the desired name in the **Filter Name**.
6. Specify 1 as **Sequence**.
7. Select the **Active** check box .
8. In the **Filter Criteria** Section, do the following:
    a. Select *Name* as a **Field**
    b. Select *starts with* as an **Operator**
    c. Specify desired category name as a **Value.** For example: Hardware.



When you make only certain categories visible through the Search Filter, Apttus CPQ displays only the categories that satisfy the criteria in the search filter on the Catalog page.

Configuring Multi-level Bundles

Your business may require you to configure a bundle in a bundle. This offers increased control over product selection, as well as making it easier to select products because selecting one bundle full of products is easier than selecting those products separately.

Depending on your business needs, you can configure products that may contain multi-level bundles. In addition to the ability to handle many levels of sub-bundles (also known as inner bundles), some of the bundles and/or options may be ramps or tiered offerings. This feature takes into account multi-level bundles with the ability to also show ramps or tiers contained within the sub-bundles. You can also apply constraint rules to multi-level bundles.

This feature enables you to configure a bundle and also view what products are being added (along with its hierarchy) to your cart on the configuration page. The selected products appear in the **My Options** section on the left hand side of the configuration page. You can also view and understand what bundle and option group you are currently in and navigate to the parent or child /lower level bundle during the configuration. The catalog/product hierarchy navigation is similar to the catalog and the configuration page. The product hierarchy is displayed in tree view and list view on the configuration page. You can minimize/hide the tree view.

To configure sub-bundles on the configuration page, you must set up the **Select Bundle Options** Visualforce page. It is recommended that after making changes to any bundle product association, you must run a bundle maintenance job that synchronizes all the bundles and options.

> Changes done in option properties like MinQuantity, IsDefault do not require you to run bundle maintenance.

To set up select bundle options page

Go to **Setup > App Setup > Develop > Custom Settings**.

1. Click **Manage** for *Config System Properties*.
2. Click **Edit** for *System Properties*.
3. In Option Page Default, type one of the following based on your requirement.

Apttus_Config2__SelectConfigOptionsDetailView	Use this when there are usage tiers, ramps and attributes for options products. This page does not support large number of options on a bundle.
Apttus_Config2__ConfigureBundle	Use this when there are no usage tiers, ramps, and attributes for option products. This page supports more than 300 options on a bundle.

4. Click **Save**.

The bundle options page is set up to display sub-bundles allowing you to configure a bundle in a bundle.

The next step is to run a bundle maintenance job.

Running a Bundle Maintenance Batch

After making changes to any bundle product association, you must run a bundle maintenance batch that synchronizes all the bundles and options.

You can also use the Update View button from the Bundle and Option Manager page to run a maintenance of view record for individual bundles. It is necessary to run a maintenance when a new bundle or bundle options are added or removed. To make sure that all bundles have view records, use the **Bundle Maintenance** tab to run a bundle maintenance batch to update view records for all bundles. The bundle maintenance tab uses the Update Bundle Components page to run the batch.

> Changes made in option properties such as MinQuanity or IsDefault do not require you to run bundle maintenance.

To run a bundle maintenance batch

1. Click and click **Bundle Maintenance**.
2. From Product Name, select the name of the parent bundle, whose sub-bundles have been changed.

3. Do one or more of the following:
   - To update the selected bundle, click Update In Batch Mode.
   - To update all the bundles in the list, click Update All.
   - To update basic bundles with a limited number of options, click Update in Fast Mode button.

   The system displays a confirmation message indicating that the process completed successfully.

The asynchronous task that maintains the product bundle structure is executed.

To run a maintenance of view record for individual bundles

1. Navigate to the **CPQ Console** tab and click **Manage Products**.

2. Select an existing bundle to which you want to add or remove a bundle or an option.
3. Click **Product Console**.
4. From the **Catalog** section, click **Manage Bundles/Options**.
5. After making changes to the product association, click **Update View**.

The view record of the selected bundle is updated.

Using a Product Family

Product Family is a standard salesforce.com feature. You can use the Product Family picklist to categorize your products. For example, if your company sells both hardware and software, you can create two Product Families: Hardware and Software.

To begin using Product Families, you must:

- Customize a Product Family picklist
- Edit a Product and select an appropriate Product Family value

Customizing the Product Family picklist

You can customize the Product Family picklist to include the different categories of products you sell.

To customize the Product Family picklist

1. Go to **Setup > App Setup > Customize > Products > Fields**.
2. From Product Standard Fields, click **Product Family**.
3. From Product Family Picklist Values, click **New**. Add one or more picklist values with each value on its own line.
4. Click **Save**.

The Product Family picklist displays the values you entered.

For each product in your price list, edit the product and select the appropriate Product Family value.

Editing a Product to include Product Family

For each product in your price list, you need to edit the product and select the appropriate Product Family value.

You must have an existing product.

To edit a product to include Product Family

1. Navigate to the **Products** tab and select a product to edit.
2. In Product Detail, from Product Family, select an appropriate value.
3. Click **Save**.

Based on your selection, your product becomes part of a Product Family.

Configuring a Product Group

Product Group (as opposed to Product Family) can consist from different product families.

In order to configure a product group, you must create a product group and add products as members of the group. You can add products as members of a group. You can also add products of different product families as a member of a product group.

To create a product group

1. Navigate to the **Product Groups** tab and click **New**.
2. In **Group Name**, enter a mandatory group name.
3. In **Description**, enter a brief description to describe the purpose of the group.
4. Click **Save**.

A product group is created and saved. After you have created a product group, you can add products as members of the group.

To add a product group member

You must have an existing product group.

1. Navigate to an existing product group and scroll down to the Products related list.

2. Click **New Product Group Member**.
3. In **Sequence**, enter a mandatory sequence number.
4. In **Product**, enter a product or click to search and select a product.
5. In **Product Group**, the product group is displayed. You can click to search and select another product group.
6. Click **Save**.

All the products added to the product group are displayed in the Products related list.

Creating Categories and Hierarchy

Categories are high-level logical groupings of products, affecting the way the end user sees them for selection on the Product Catalog. A category can be created for browsing products in the Catalog page or they can be created for creating options groups for a Bundle product. Products are associated to a category through a category hierarchy. Category hierarchies are maintained using the Hierarchy Manager. They are also used to total the product prices into logical groupings.

You can edit existing categories based on your specific business requirement.

> **Note**
>
> Making changes to existing categories or category hierarchies requires you to run the Category Maintenance batch job to maintain the hierarchy and associations.

- Creating Categories
  Categories are created to group products for the end user's product selection. Categories are also used to group options within bundles.
- Working with Hierarchy Manager
  Categories are created when a hierarchy needs to be created for the first time. Hierarchy defines the structure of the product catalog.
- Running a Category Maintenance Job
  After making any change to category definition, category to product association, or hierarchy change, you must run a Category Maintenance job. This de-normalizes the hierarchy into a custom object for reporting and totaling purposes. If category maintenance is not run, the end user may see incorrect hierarchy or totals on the cart page.

Creating Categories

The purpose of categories is to group products for the end user’s product selection. Categories are also used to group options within bundles.

There are two types of category:

- Offering: An Offering is a collection of products that an organization makes available to a customer. A Bundle or a Standalone product can be associated with an Offering category. If you are creating a category hierarchy for the first time, you need to create a new category via this method only. Most implementations will have an existing hierarchy.
- Option Group: Option Groups categorize multiple option products. Option products are associated to option groups, and option groups are further associated to bundle products, thus associating option to a bundle product.
- Both: This category type groups products that are of type Standalone and Option. For example: A keyboard can be sold as a standalone product and also as an option along with a bundle product (such as Desktop PC).

To create a category

From the Salesforce.com application menu, select **Apttus Product Setup**.

1. Go to the **CPQ Console** tab, and from the Product Management section, click **Add Categories**.
2. Type a mandatory **Category Name**, and a mandatory **Category Label**.
3. To make the category active, select **Active**. By default, the **Active** check box is already selected.
4. From the Type picklist, select one of the following:
    - Offering - Select this to categorize Standalone and Bundle products. This is the top level category.
    - Option Group - Select this to group Option products.
    - Both - Select this to group products that are Standalone and Option products.

5. Click **Save**.

Based on your category type selection; an Offering, an Option Group or Both type category is created and added.

Create categories based on your business requirement and start building a hierarchy using the hierarchy manager.

To edit an existing category

1. From the Salesforce.com application menu, select **Apttus Product Setup**.
2. Go to the **CPQ Console** tab, and from the Product Management section, click **Manage Categories**.
3. Select an existing category to change and click **Edit**.
4. In Category Name and **Category Label**, you can change the existing category name and category label.
5. To make the category active, select the Active check box.
6. From the **Type** list, select a category type.
7. Click **Save**.

The category is modified and saved. After modifying the existing categories, you must run the Category Maintenance batch job.

Sequencing Products within a Category

You can use the orderByClause parameter to sequence the Products on the Catalog page. You can add a new field in the Product2 object (for example: as Category Rank) and set the orderByClause in this manner: productOrderByClause=Category_Rank__C NULLS LAST, Name while configuring a formula to launch the Product Catalog.

By default, the product list shown on the Catalog page is sorted by **Product Name**.

To sequence the Products on the Catalog page

1. Create a new custom field on Product2 object. For example: Category Rank.
2. From the Product Detail screen, assign the appropriate numeric value for this filed.
3. Use the following formula to launch the ngCart.

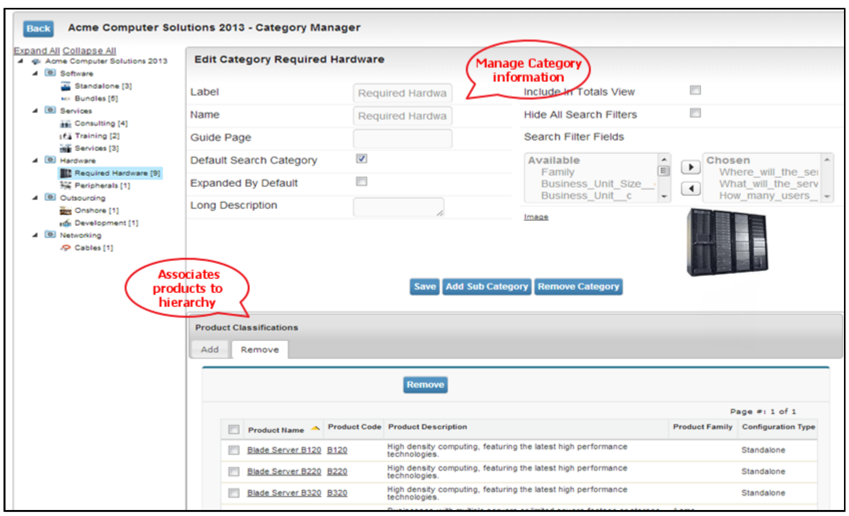

```
IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 ,HYPERLINK("/apex
/Apttus_QPConfig__ProposalConfiguration?id=" &Id&"&flow=ngFlow&produc
tOrderByClause=Category_Rank__c NULLS LAST, Name" ,IMAGE("/resource
/Apttus_QPConfig__Button_Configure", "Configure Products"),"_self"),
NULL)
```

On the Catalog page, Products are displayed as per the sort order defined in the custom field.

Working with Hierarchy Manager

Category hierarchy defines the structure of the product catalog. Use Hierarchy Manager to build and maintain a category hierarchy. You can add sub-categories (also known as nodes) to a category, add an image, reorder categories, and associate products, thus building a hierarchy. Most implementations will have an existing hierarchy.

You can create multi-level categories. For example: **Home > Laptops > Model**.



To build a hierarchy

You must have an existing Offering category.

1. Go to the CPQ Console tab, and from the Product Management section, click **Manage Categories**.
2. Select an existing category and click **Hierarchy Manager**. The Category Manager page is displayed, which allows you to perform the following actions for the selected category.
   - Add or Edit a Sub category
   - Remove a category

- Add an image
- Associate Products
- Reorder Categories

3. Click **Save**.

A category hierarchy is built. The hierarchy is visible on the left hand side of the Category Manager page.After building the hierarchy, you must run the Category Maintenance batch job.

- Adding a sub-category
  Sub categories (also known as nodes), categorize option products within an option group. You can add multiple sub-categories to categorize option products within an option group.
- Associating a Product to a Category from the Hierarchy Manager
  You can use the Hierarchy Manager to associate a product to a category or sub-category and also to build a hierarchy. You can also associate a product to a category from the Product Console.
- Associating an image
  An image can be associated to a category, sub-category or a product for visual clarity. You can add images of three sizes.
- Removing a sub-category
  You can remove a sub category and its nested categories from a hierarchy at the same time. The Remove Category button is available only for a lower level node.
- Re-ordering categories
  You can rearrange the order of your categories or sub-categories within a hierarchy. Reordering of categories can be done from the leaf nodes. Leaf nodes are the level of categories between the top level category and the lowest node.
- Using Search Filter to find products
  Apttus CPQ has a built-in comprehensive search feature. Use keyword search for a product or use the Advanced Options search feature to search for a product using filter criteria.

Adding a Sub category

Sub categories (also known as nodes), categorize option products within an option group. You can add multiple sub-categories/nodes to categorize option products within an option group.

You must have an existing top level category. You must be on the category manager page.

To add a sub-category

1. Select an existing category or sub-category from the hierarchy, and click **Add Sub Category**.

   > **Note**
   >
   > You cannot add a sub-category to a category that has products associated to it.

2. Type a Label and a **Name** for the sub-category.
3. In the **Guide Page**, type the guided selling Visualforce page name.
4. To display products in a category on the catalog page, select **Default Search Category**.

   When checked, Default Search Category displays the product by default under the category /sub-category in the catalog page. If unchecked, the user has to first select the category from the category panel or the Browse section to see the product.
5. To keep the hierarchy expanded at all times in the catalog page, select **Expanded By Default**.
6. Type a **Long Description**, and to display the node in the totals view on the shopping cart, select **Include In Totals View**. The Long Description is displayed for categories that have bundle products on the Options page.
7. To hide all the search filters on the catalog page in Narrow Your Search and Refine Your Search sections, select Hide All Search Filters.
8. In the Search Filter Fields, from the Available list, select a field and click the **Add** button to move it to the Chosen list to display the filter fields in the Refine Your Search section.
9. To add a small image like a symbol or icon for the sub-category, click **Image**.
10. To add a larger image for the sub-category, click **Large Image**.
11. Click **Save**.

The sub-category is added as a node to the hierarchy.

Associating a Product to a Category from the Hierarchy Manager

You can use the Hierarchy Manager to associate a product to a category and also to build a hierarchy. You can also associate a product to a category from the Product Console.

To associate a product to a sub-category

Navigate to the category that you want a product to be associated with.

1. From the Product detail page, click **Hierarchy Manager**.
2. From the hierarchy, select the lowest level node.
3. From the **Product Classifications** section, from the list of available products,select a product that you want to associate and click Associate. You may also search for a product using the search filter criteria. All the associated products are displayed in the **Remove** tab of the Product Classifications section.
4. Click **Save**.

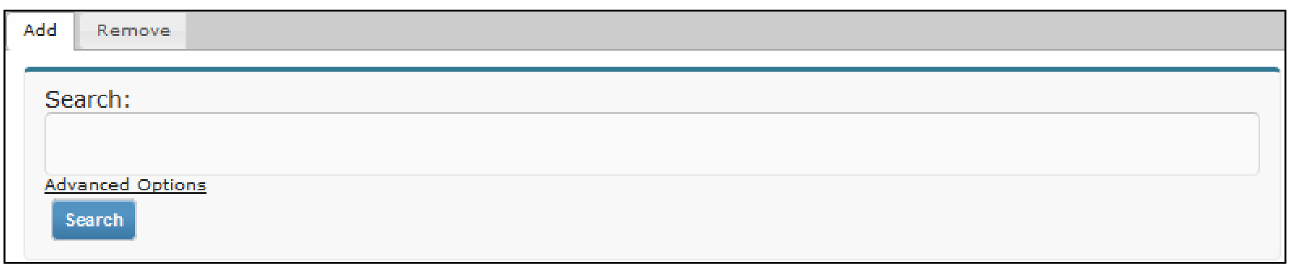
Your desired product is associated with a category.

Using the Search Filter

Apttus CPQ has a built-in comprehensive search feature. Use keyword search for a product or use the Advanced Options search feature to search for a product using filter criteria.

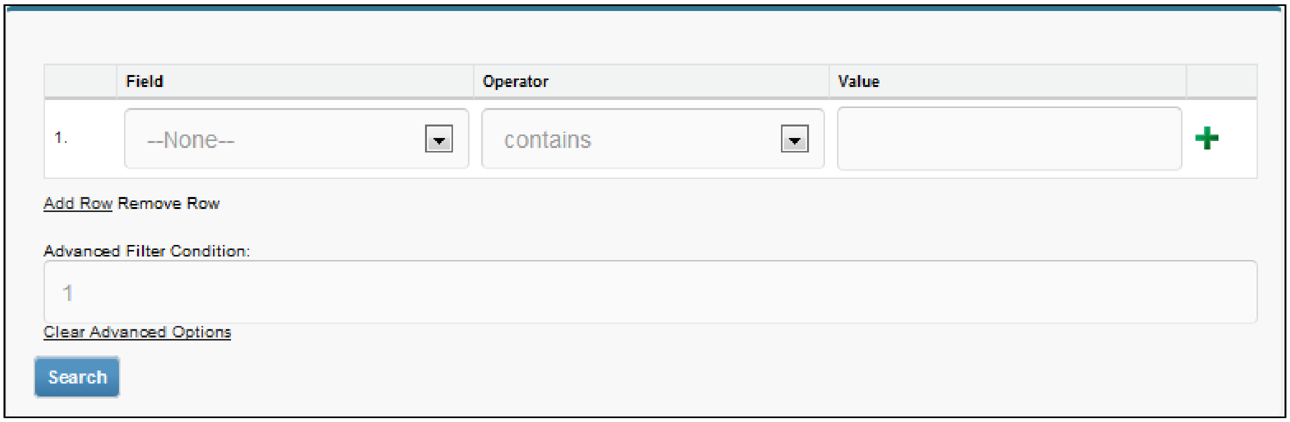
Standard Search:

You can search for a product using a product name.



Advanced Options

You can use advanced search to search for a product based on various filter criteria.



Associating an Image

An image can be associated to a category, sub-category or a product for visual clarity. You can add images of three sizes depending on the image placement on the Catalog page.

The image sizes are small, medium and large. To associate a small image, you can use a 16X16 pixel size image, for a medium image 50X50 pixel, and for a large image, the size is 100X100 pixels.

> You must be in an existing category to which you want to associate an image.

To associate an image

1. Click the Image or Large Image link.
2. Enter the path of the File, click **Browse**. Browse to the image file, select it and click Open.
3. Click on the "Attach Icon" button, click **Attach Icon**. After the file is completely uploaded, the image appears in the preview pane.
4. Click the "Done" button to return to the previous page. If you click Done during an in-progress upload, the upload is cancelled.

An image is associated with a category.

Removing a Sub-category

You can remove a sub-category and its nested categories from a hierarchy at the same time. The **Remove Category** button is available only for a lower level node.

To remove a sub-category

1. Select the sub-category from the hierarchy that you want to delete and click **Remove Category**.
2. Click **OK** to remove the category and any of its nested categories.

Sub-category and all its nested categories are removed from the hierarchy.

Reordering Root Categories on the Catalog Page

Product managers can correctly sequence the root or top level catalog categories to ensure that the right product lines are prioritized or that the most common categories are positioned first. The ordering can be different for different Price Lists. Price Lists drive the hierarchy that shows up on the catalog for a given quote.

To reorder or sequence the root categories (top level categories on the catalog page)

The categories are associated with the category hierarchy and the category hierarchy is associated to a Price List for it to show up in the catalog.

1. Go to your target Price List. Navigate to the CPQ Console and click Manage Price Lists.
2. Under the related list for Categories, edit the target category hierarchy that is related to your target root (top level) category.
3. Set the sequence number relative to other categories to drive the order (Object: Price List Category, Field: Sequence).
4. Click Save.

The order set by the sequence number for a given Price List should be the order in which the categories show up on the catalog page. You do not need to run the maintenance job after updating the sequence.

You also can rearrange the order of your categories or sub-categories within a hierarchy. Reordering of categories can be done from the leaf as well as root nodes. Leaf nodes are the level of categories between the top level category and the lowest node.

To re-order a sub-category

1. Select a leaf, root node/category from the hierarchy and click **Reorder**.
2. Click or next to the category to move the category up or down.
3. After you have finalized the order, click .

The system updates the order and displays the hierarchy.

Running a Category Maintenance Job

After making any change to category definition, category to product association, or hierarchy change, you must run a Category Maintenance job. This de-normalizes the hierarchy into a custom object for reporting and totaling purposes. If category maintenance is not run, the end user may see incorrect hierarchy or totals on the cart page.

IMPORTANT

You must run the Category Maintenance batch job after every CPQ version upgrade.

To run a category maintenance job

Pre-requisite: Go to **Setup > Administration Setup > Email Administration > Deliverability**. In the **Access to Send Email** section, ensure that the **Access Level** is set to **All Email**.

1. Navigate to the **Category Maintenance** tab.
2. From the Hierarchy picklist, select the top-level category for your organization. Typically this will be the name of your organization.
3. Click **Update View**.

This executes an asynchronous batch job that maintains the hierarchy. After you click Update View, the administration task is complete and an updated history for all the batch jobs is displayed.

Cloning a Bundle within the same Quote

You can clone a bundle within the same quote, make modifications, and use it as another bundle.

This feature enables you to clone a bundle with attributes, options, and pricing, make desired changes and save it as another bundle.

To clone a bundle within the same quote

You must have an existing bundle.

1. Select a bundle and click **Validate**. You can only clone a bundle.
2. Click  (Copy icon).
3. You can configure the bundle or make the desired changes and click **Validate**.

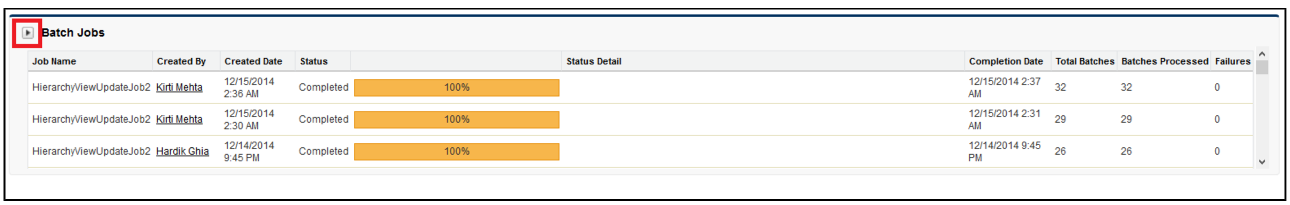
You have created another bundle from an existing bundle.

You can proceed to pricing.

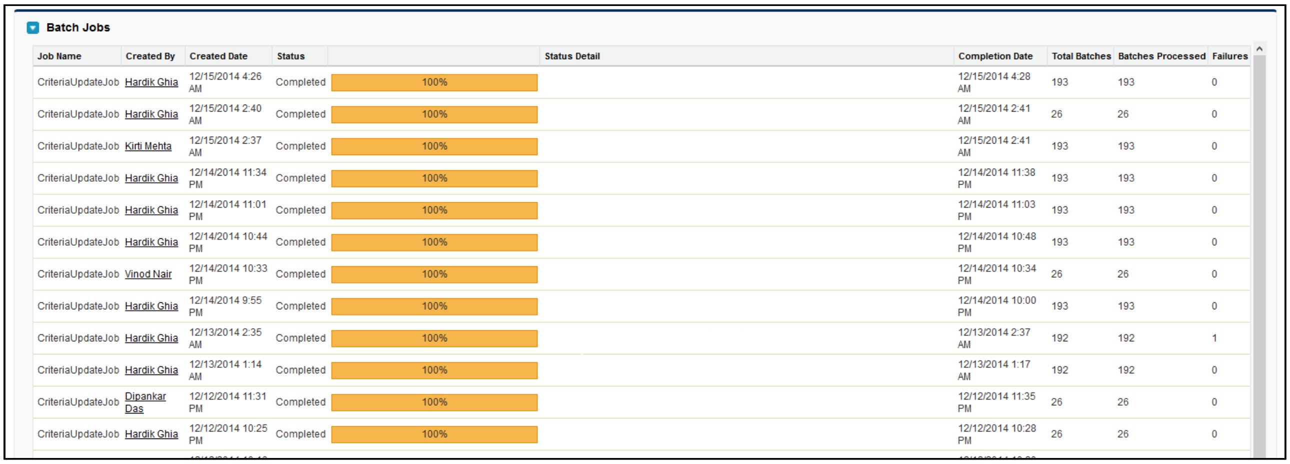
Expanding the Batch Jobs Section

The Batch Jobs section can be expanded to display all the batch jobs run.

Previously, when a maintenance task was run, such as *Criteria Maintenance*, *Category Maintenance* and more, a few entries were listed with a scroll bar to navigate to see the entire list.



When you click the expander icon  , the *Batch Jobs* section expands to display more of the batch jobs.



Batch Jobs

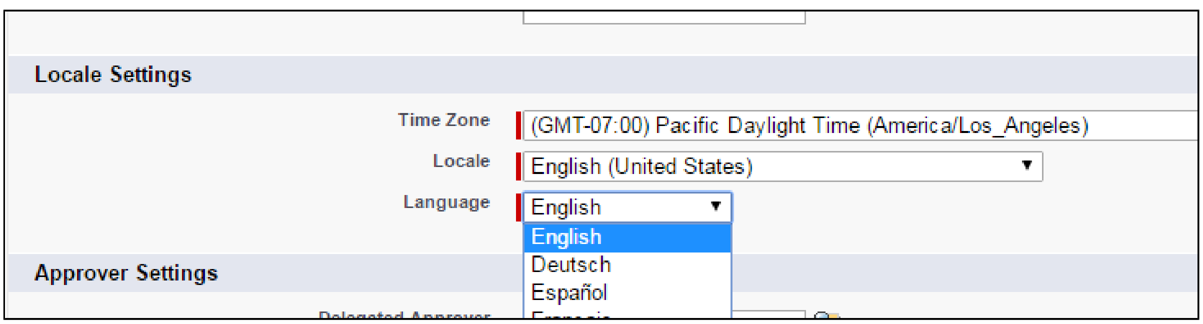
Job Name	Created By	Created Date	Status		Status Detail	Completion Date	Total Batches	Batches Processed	Failures
CriteriaUpdateJob	Hardik Ghia	12/15/2014 4:26 AM	Completed	100%		12/15/2014 4:28 AM	193	193	0
CriteriaUpdateJob	Hardik Ghia	12/15/2014 2:40 AM	Completed	100%		12/15/2014 2:41 AM	26	26	0
CriteriaUpdateJob	Kirti Mehta	12/15/2014 2:37 AM	Completed	100%		12/15/2014 2:41 AM	193	193	0
CriteriaUpdateJob	Hardik Ghia	12/14/2014 11:34 PM	Completed	100%		12/14/2014 11:38 PM	193	193	0
CriteriaUpdateJob	Hardik Ghia	12/14/2014 11:01 PM	Completed	100%		12/14/2014 11:03 PM	193	193	0
CriteriaUpdateJob	Hardik Ghia	12/14/2014 10:44 PM	Completed	100%		12/14/2014 10:48 PM	193	193	0
CriteriaUpdateJob	Vinod Nair	12/14/2014 10:33 PM	Completed	100%		12/14/2014 10:34 PM	26	26	0
CriteriaUpdateJob	Hardik Ghia	12/14/2014 9:55 PM	Completed	100%		12/14/2014 10:00 PM	193	193	0
CriteriaUpdateJob	Hardik Ghia	12/13/2014 2:35 AM	Completed	100%		12/13/2014 2:37 AM	192	192	1
CriteriaUpdateJob	Hardik Ghia	12/13/2014 1:14 AM	Completed	100%		12/13/2014 1:17 AM	192	192	0
CriteriaUpdateJob	Dipankar Das	12/12/2014 11:31 PM	Completed	100%		12/12/2014 11:35 PM	26	26	0
CriteriaUpdateJob	Hardik Ghia	12/12/2014 10:25 PM	Completed	100%		12/12/2014 10:28 PM	26	26	0

Multi Language Support for CPQ

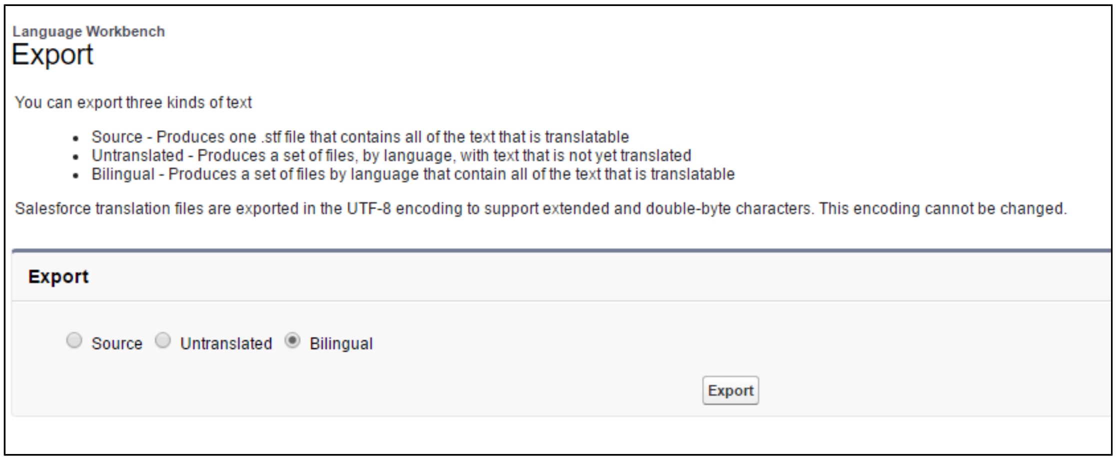
Using this feature, you can display translated fields when the user belongs to a different locale. Multi Language support is available for Apttus out of the box field names, picklist values, button labels, and error messages.

Configurations for translating content in another language

- To use Translation Workbench for translating the label data, go to **Setup > Administer > Translation Workbench > Translation Settings**. Click **Add** button and select the **Language** as *Japanese*. To change the language at the Profile level, go to **Setup > Users > Edit**. Under the **Locale Settings**, change the value in the Language field and click **Save**.



- To add translations for custom labels (i.e non-Apttus out of the box labels), you need to upload a new SalesForce Translation File (STF) via translation workbench.
  - Export Bilingual STF from Setup > Translation Workbench > Export



- Translate STF (non-translated items will show up in English).
- Import Translation from Setup > Translation Workbench > Import.

Creating and Maintaining Attributes

Attributes are used to associate certain features to a product. Examples of common attributes are color, size, weight, and more. Attributes are associated to a product either to drive pricing or simply stored for informational purposes against the line item for other purposes. Attributes are associated to products through Attribute Groups.

The Attributes administration page allows you to create and delete attribute groups that can be associated or disassociated with products. You can manage attribute groups and attributes associated to a product. You can also manage the sequence for associated attribute groups.

The subsequent chapters in this page explain the following functionalities:

- Creating a Product Attribute field
  Product Attributes are used to associate certain features to a product. You must create product attribute fields in the Product Attribute Value object to represent an attribute. These attribute fields are then associated to a product via attribute groups.
- Creating a Product Attribute Group
  Attribute groups are groupings of Product Attributes. You must create a product attribute group and associate product attribute field to it. Multiple product attributes can be grouped together.

- Associating an Attribute Group to a Product
  You must associate an attribute group to a product for displaying the attributes (features) on the attributes page.
- Setting default Price from Attribute using a Formula field
  You can add a formula field in the Product Attribute Value object and use it in a Price List Item to derive the Default Price from the attribute. In this way, you no longer have to add the attribute to the attribute group and associate it with the product.
- Configuring Numeric Expressions for Attributes
  You can set the default price for an attribute using a numeric expression.

Creating a Product Attribute field

Product Attributes are used to associate certain features to a product. You must create product attribute fields in the Product Attribute Value object to represent an attribute. These attribute fields are then associated to a product via attribute groups.

To create a product attribute field

Go to **Setup > App Setup > Create > Objects**.

1. On the Custom Objects page, scroll down and select the **Product Attribute Value** object.
2. Scroll down to the Custom Fields & Relationships related list, and click **New**. You will be directed to the standard Salesforce field creation wizard. With the help of the wizard, you can create a product attribute field, set up field dependencies, define workflow rules, line item validation rules, defaults, and more.

The Product Attribute field is created.

The next step is to create an Attribute Group and associate the product attribute field you created to the attribute group.

Creating a Product Attribute Group

Attribute groups are groupings of Product Attributes. Product Attribute fields must be associated to a Product Attribute group, which is to be associated to a product. Multiple product attributes can be grouped in a single Product Attribute group and a product can have multiple Product attribute groups associated to it.

To create a Product Attribute Group

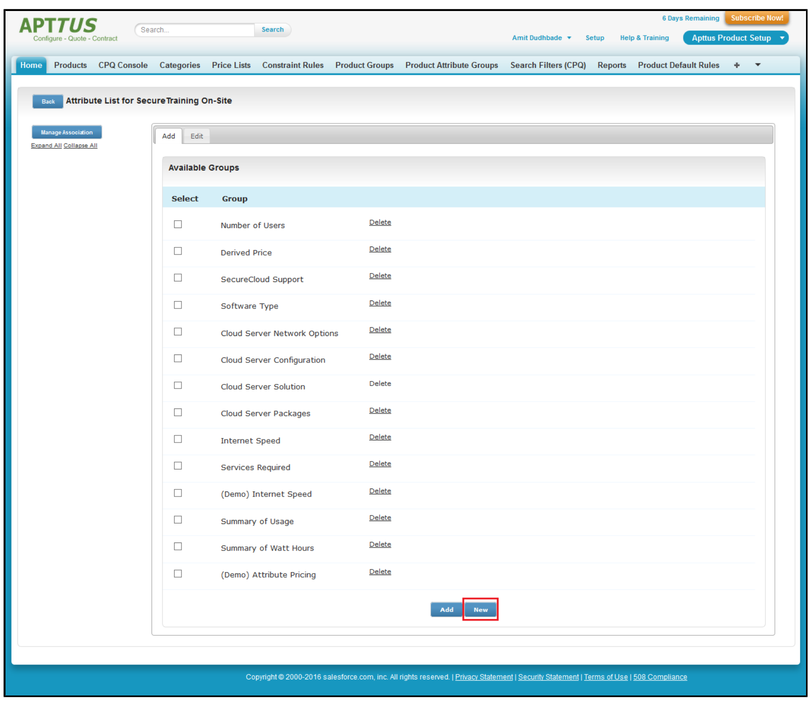
Choose any of the following ways to create a new Product Attribute Group:

1. Go to **All Tabs** and click **Product Attribute Groups**.
2. Enter a mandatory **Group Name** and select the **Business Object** for your attributes. The available values are *Product Attribute Value* and *Search Attribute Value.*
3. Depending on your Business Object, different attribute fields are populated in the **Field.**
4. Select **Is Read Only** if you do not want the end user to change the value of an attribute. Select **Is Hidden** if you want to associate an attribute to your product but do not want to display the attribute for selection.

   Use to toggle the order of the attributes.
5. Click **Save**.

-- OR --

1. Navigate to the **Products** tab and select a bundle product.
2. Click **Product Console** and from the Additional Data section, click **Attributes**.
3. Click **Manage Association** and click **New**.



4. Type a mandatory **Group Name**, and type a Description.
5. To display the attributes in two columns, select **Two Column Attribute Display**.
6. In the Product Attribute Edit section, to make an attribute field read-only, select **Is Read Only**
   .
7. In the **Field** column, type in the attribute name and select the attribute field that you want to associate with the attribute group. More than one attribute can be associated to a group and thus shared across groups. You can do one or more of the following:
   - To add a new row, click **Add**.
   - To delete a row, click **Del**.
   - To reorder an entry, click  or  next to it.
   - To create and add a new attribute field, click **Create Field**.
8. Click **Save**.

Associating an Attribute Group to a Product

You must associate an attribute group to a product for displaying the attributes (features) on the attributes page.

To associate a product attribute group to a product

You must have an existing product attribute group with product attribute field associated to it.

1. Navigate to the **Products** tab and select a bundle product.
2. Under Product Details, select the **Has Attributes** check box. Clearing the **Has Attributes** check box disables the attributes page.
3. Click **Product Console** and from the Additional Data section, click **Attributes**.
4. Click **Manage Association**.
5. From the Available Groups list, select one or more attribute group and click **Add**.

An attribute group is associated to a product.

Setting default Price from Attribute using a Formula field

You can add a formula field in the Product Attribute Value object and use it in a Price List Item to derive the Default Price from the attribute. On completing this, you no longer have to add the attribute to the attribute group and associate it with the product.

This way, you can make sure that this attribute doesn't show up in the configuration pages when the end user is going through the configuration process.

Attribute-Based Configuration

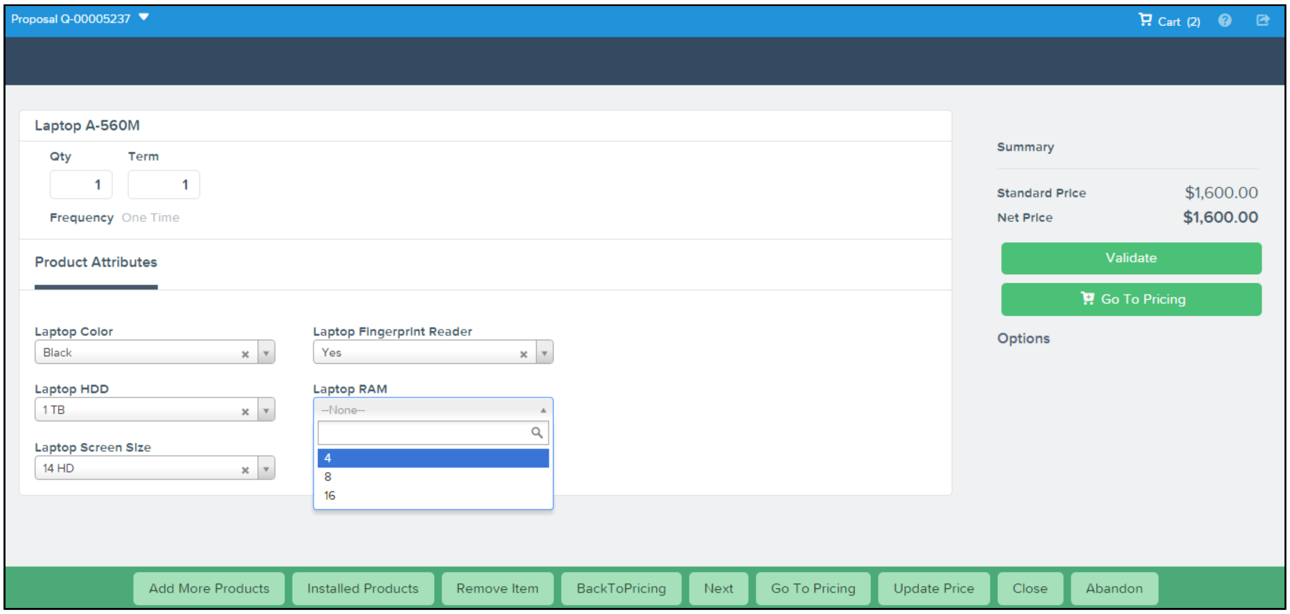
Attribute-Based Configuration helps you customize the product configuration on your cart on the basis of selection of product attributes. The attributes define characteristics or the desired features for the product. The attribute configuration may or may not affect price of the product. Each attribute value selection ensures that the rest of the attributes get filtered on the basis of selected attributes and the result set is narrowed. The attributes can also leverage expressions and calculations to derive values that help identify the products.

Note

- Ensure that you have selected **Has Attributes** check box in your product.
- Ensure that you have selected **Enable Field Expression** in the Config System Properties.

With the New UI, Apttus CPQ allows you to create and configure unlimited number of attributes, overcoming the 800 record limit of Salesforce. You can use additional attributes for Product Definitions, Product Configurations and Attribute-based Rules.

For an example, suppose you have a product configuration behavior similar to a laptop review website where the selection of RAM and HDD restricts the Color and Screen Size, and selection of Color and Screen Size restricts any other component. The dependencies can be in any direction without having a set selection sequence. Within Apttus, you have to create a Compatibility table to define all the valid combinations for the Color, Screen Size, RAM, HDD, and Fingerprint Reader picklist values.



Attribute-based configuration provides a rules framework to drive configuration based on your Product Attributes. The rules framework allows you to set up the following dependencies or rules between the product attributes. You should perform the following steps depending on your business requirements.

1. Create Products
2. Create Product Attributes
3. Create a Product Attribute Group
4. Associate Product Attribute Group to a Product
5. Create Product Attribute Values
6. Create Attributes in Attribute Value Matrix Entry object
7. Create Attribute Value Matrices
8. Create Product Attribute Rules

Creating Product Attribute Values

You must create the attributes that you want to add in your product as custom fields in the Product Attribute Value object.

In our example, we have added Laptop Color, Laptop RAM, Laptop Screen Size, Laptop Fingerprint Reader, and Laptop HDD attributes to the Product Attribute Value object. Hence, we must add these attributes in the Attribute Value Matrix Entry object as well.

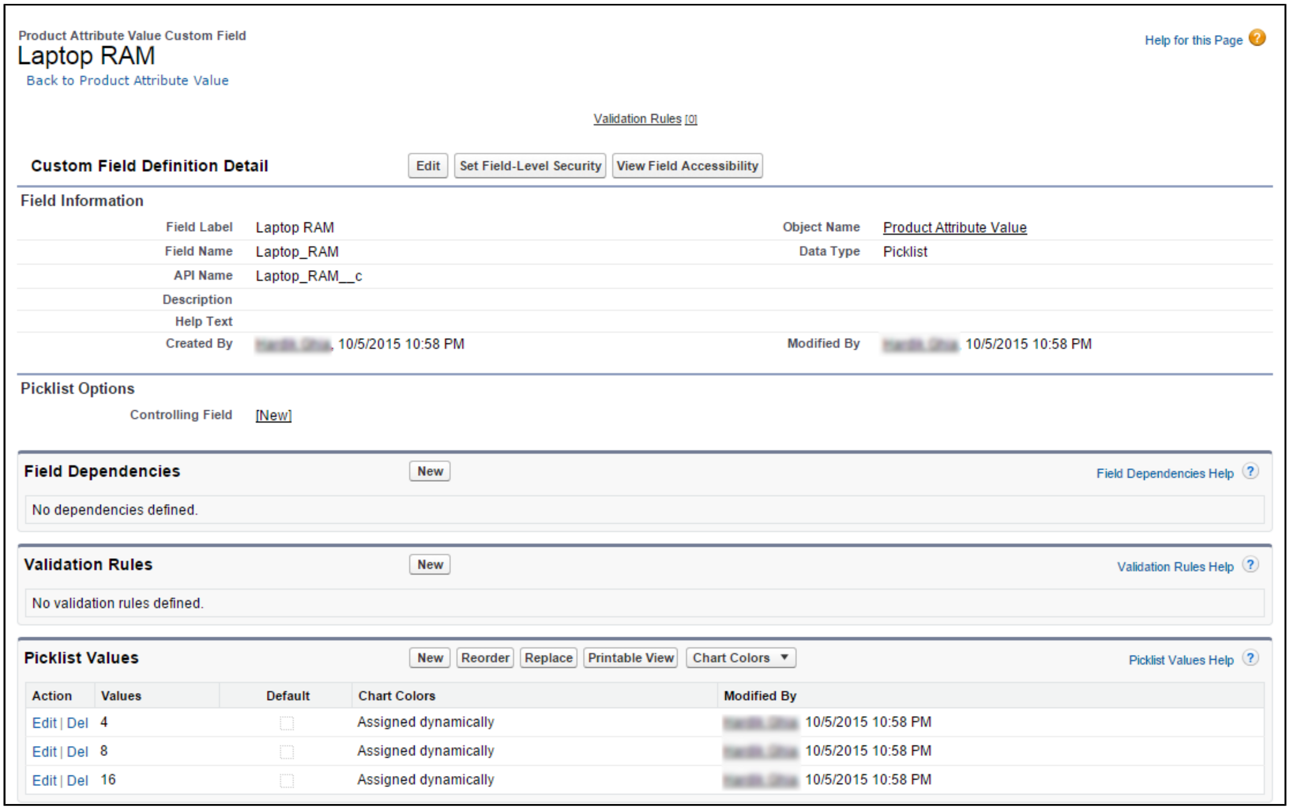
To create product attribute value

1. Go to **Setup > App Setup > Create > Objects** and select **Product Attribute Value**.
2. Under Custom Fields & Relationships section, click **New**.
3. Select **Picklist** from the Field Type.
4. Click **Next**.
5. Enter **Field Label**.
6. Enter values for picklist.
7. Click **Next**.
8. Establish field level security and click **Next**.
9. Select appropriate page layouts where you want to add this field.
10. Click **Save.**

Let's create Laptop RAM attribute in the Product Attribute Value object.

1. Go to **Setup > App Setup > Create > Objects** and select **Product Attribute Value**.
2. Under Custom Fields & Relationships section, click **New**.
3. Select **Picklist** from the Field Type.
4. Click **Next**.
5. Enter Laptop RAM as **Field Label**.
6. Enter 4, 8, 16 (separated by a line) as values for picklist.
7. Click **Next**.
8. Establish field level security and click **Next**.

9. Select appropriate page layouts where you want to add this field.
10. Click **Save.**



Similarly, create Laptop Color, Laptop HDD, Laptop Screen Size, and Laptop Fingerprint Reader attributes in the Product Attribute Value object.

You have created attributes in Product Attribute Value object.

Now, you must create attributes in Attribute Value Matrix Entry object.

Creating Product Attribute Value Extensions

Apttus CPQ has been enhanced to create and configure an unlimited number of attributes, overcoming the 800 record limit of Salesforce. The additional attributes can be used for Product Definitions, Product Configurations and Attribute-based Rules.

These attributes are created as an extension to the **Product Attribute Value** object through a combination of master-detail and look-up relationships. The extension objects are identical to the existing **Product Attribute Value** object.

For every single record in the base **Product Attribute Value** object, the extension objects will also have a single record under them, thus increasing the number of attributes a product can have. All of these attributes created under the extension object can be used as normal attributes for your products.

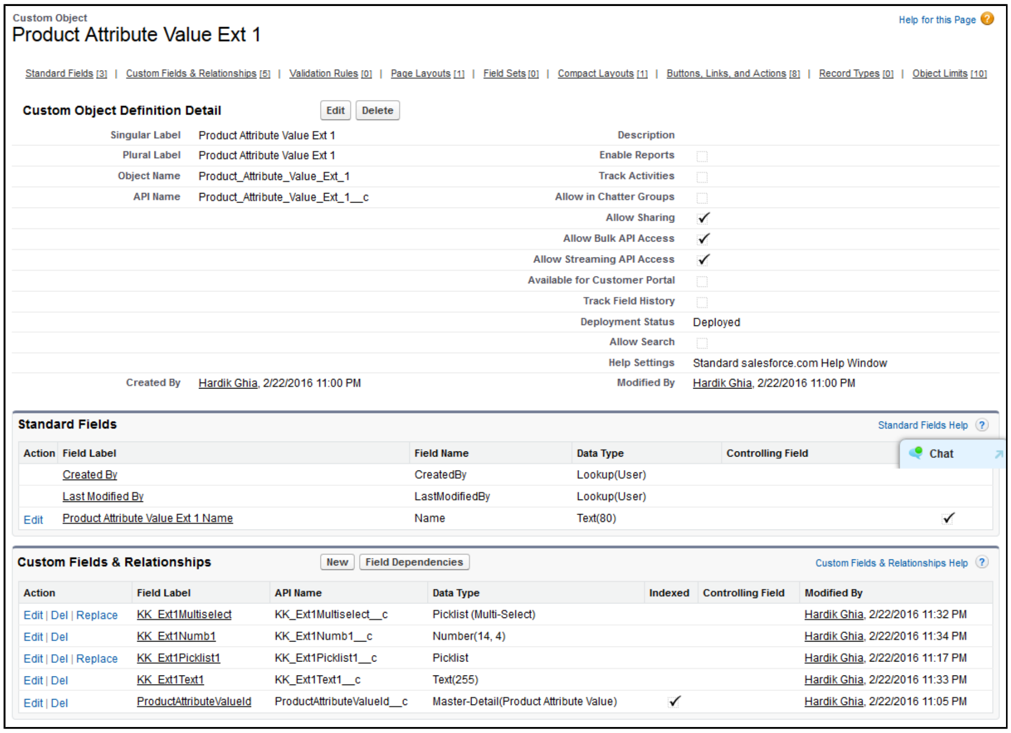
The fields from the extension objects are displayed in the Product Console to be associated as attributes with the products.

- This enhancement is available in the New UI only.
- The unlimited attributes are not available for Price Dimensions (Pricing engine). Only the first 800 attributes will be visible for the pricing.

To create an extension object

1. Navigate to **Setup > App Setup > Create > Objects** and click **New Custom Object** to create a new extension object.
2. Enter a desired label, such as *Product Attribute Value Ext 1* to indicate that this object is an extension to **Product Attribute Value** object.
3. Fill in the requisite details and click **Save**.
4. Navigate to the Custom Fields & Relationships section and click **New** to create a new custom relationship.
5. Choose *Master-Detail Relationship* as the Field Type and click **Next**.
6. From the **Related To** drop-down list, choose *Product Attribute Value* and click **Next**.
7. In the **Field Label**, type *ProductAttributeValueId* and click **Next**.
8. Choose necessary Field level security for the reference field and click **Next**.
9. Choose the page layouts where you want to reference this field and click **Next**.

10. Verify the Related List label for your field and click **Save**. Ensure that the API name for the newly created custom relationship is *ProductAttributeValueId__c* , as shown in the figure below.



After you have created the extension object, follow the same steps mentioned here to create the attributes for this object in the form of custom fields.

To create a lookup relationship with the Product Attribute Value extension object

For each extension object , a lookup relationship must be created so that the objects are connected.

1. Navigate to **Setup > App Setup > Create > Objects** and search for **Product Attribute Value** object.
2. Under the Custom Fields & Relationships section and click **New** to create a new custom relationship.
3. Choose *Lookup Relationship* as the Field Type and click **Next**.

4. From the **Related To** drop-down list, choose the object you created in the above procedure. In our example, choose *Product Attribute Value Ext 1* and click **Next**.
5. In the **Field Label**, type *ProductAttributeValueId*. In the **Field Name**, type *ProductAttributeValueId1* and click **Next**.
6. Choose necessary Field level security for the reference field and click **Next**.
7. Choose the page layouts where you want to reference this field and click **Next**.
8. Verify the Related List label for your field and click **Save**.
   Ensure that the API name for the newly created custom relationship is *ProductAttributeValueId1__c.*

Field Information

Field Label	ProductAttributeValueId	Object Name	Product Attribute Value
Field Name	ProductAttributeValueId1	Data Type	Lookup
API Name	ProductAttributeValueId1__c		
Description			
Help Text			
Created By	Hardik Ghia, 2/22/2016 11:11 PM	Modified By	Hardik Ghia, 2/22/2016 11:11 PM

Lookup Options

Related To	Product Attribute Value Ext 1	Child Relationship Name	Product_Attribute_Values
Related List Label	Product Attribute Values		
Required			
What to do if the lookup record is deleted?	Clear the value of this field.		

To configure Custom Settings for the extension object

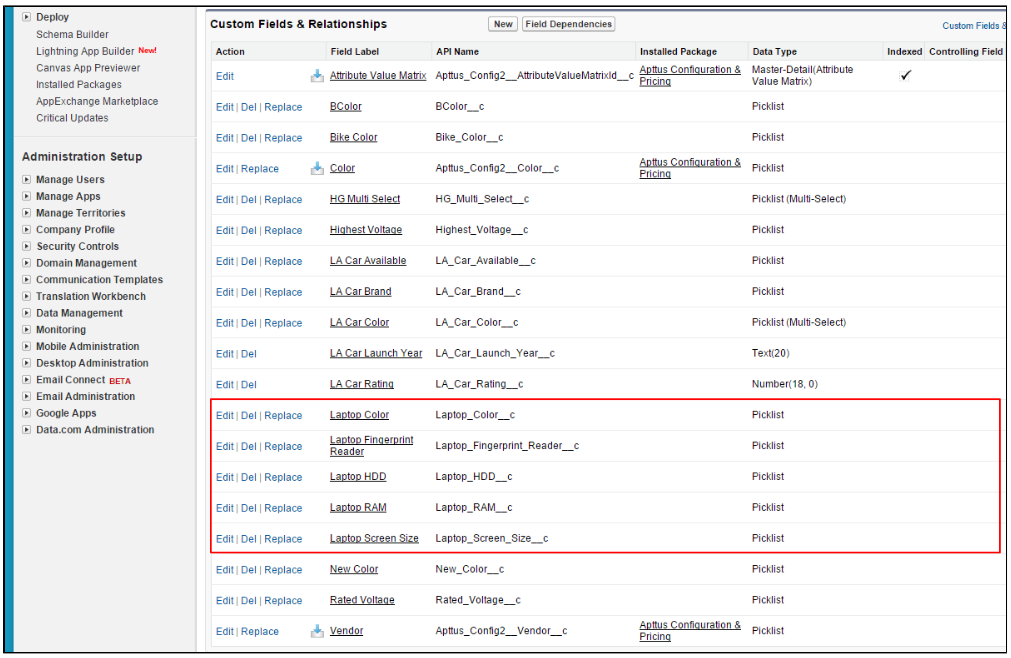
For each lookup relationship you create with the extension object, you must make an entry in a custom setting.

1. Navigate to **Setup > App Setup > Develop > Custom Settings** and click **Manage** next to **Config System Properties**.
2. Next to **System Properties**, click **Edit**.
3. In the **Product Attribute Extension Tables** field, type the API name of the lookup field you have created in the above procedure. In our example, type *ProductAttributeValueId1__c* . Ensure that you have specified the API names of all your lookup relationships here, separated by a comma. For example, if you have 4 extension objects and have their corresponding 4 lookup relationships, you must specify 4 API names in this field.
4. Click **Save.**

Creating Attributes in the Attribute Value Matrix Entry object

The Attribute Value Matrix Entry object must comprise attributes that are the same as the entries in Product Attribute Value with the same API Name. The attributes that you add in the Attribute Value Matrix Entry object are displayed on the Attributes page on the cart.

In our example, we have added Laptop Color, Laptop RAM, Laptop Screen Size, Laptop Fingerprint Reader, and Laptop HDD attributes to the Product Attribute Value object. Hence, we must add these attributes in the Attribute Value Matrix Entry object as well.



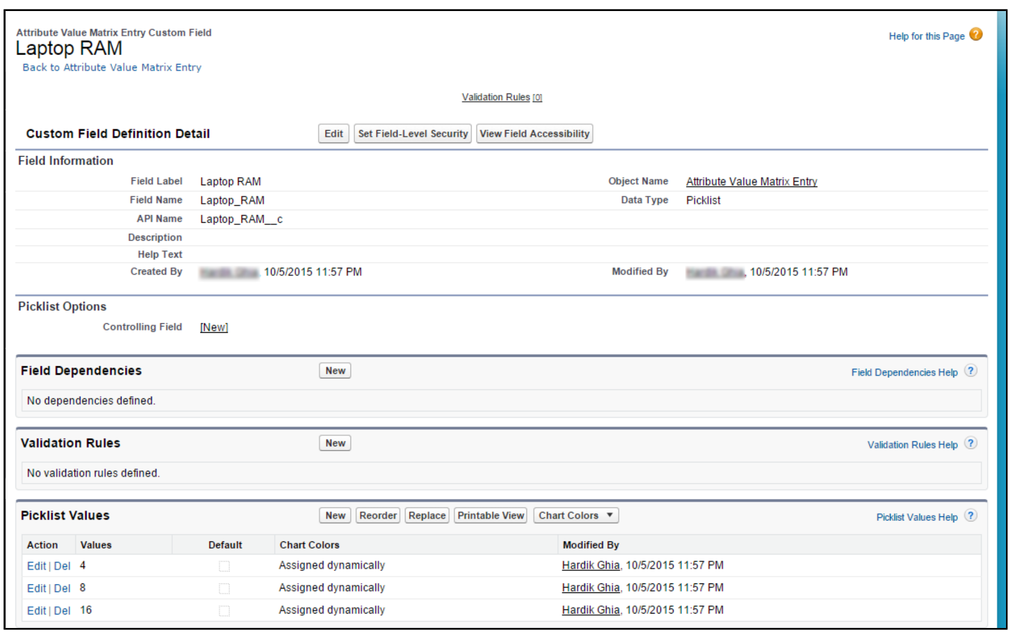
To create an attribute in the Attribute Value Matrix Entry object

1. Go to **Setup > App Setup > Create > Objects** and select **Attribute Value Matrix Entry**.
2. Under Custom Fields & Relationships section, click **New**.
3. Select **Picklist** from the Field Type.
4. Click **Next**.
5. Enter **Field Label**.
6. Enter values for picklist.
7. Click **Next**.
8. Establish field level security and click **Next**.

9. Select appropriate page layouts where you want to add this field.
10. Click **Save.**

Let's create Laptop RAM attribute in attribute value matrix entry object.

1. Go to **Setup > App Setup > Create > Objects** and select **Attribute Value Matrix Entry**.
2. Under Custom Fields & Relationships section, click **New**.
3. Select **Picklist** from the Field Type.
4. Click **Next**.
5. Enter Laptop RAM as **Field Label**.
6. Enter 4, 8, 16 (separated by a line) as values for picklist.
7. Click **Next**.
8. Establish field level security and click **Next**.
9. Select appropriate page layouts where you want to add this field.
10. Click **Save.**



Similarly, create Laptop Color, Laptop HDD, Laptop Screen Size, and Laptop Fingerprint Reader attributes in the Attribute Value Matrix Entry object.

You created attributes in the Attribute Value Matrix Entry object.

Now, you should create the Attribute Value Matrix.

Creating Attribute Value Matrices

Attribute Value Matrices are used to associate attributes to the product. You can use the attribute value matrices to drive the product selection. You can associate a maximum of five attributes with each other. The attributes defined in the Attribute Value Matrices are bi-directional. Once you create a matrix, a compatibility table is created. This compatibility table for Attribute Values helps you define all the valid combinations for the attribute values.

For example, Amttus, a leading laptop manufacturer, wants to use attributes to drive the product selection of their users. A laptop has five attributes: Color, HDD, RAM, Screen Size, and Fingerprint Reader. The following table describes the products and its attributes.

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB
RAM	8	16	4	8	8	16
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

After you create an attribute value matrix, when the user selects Black from the Color picklist menu, the HDD picklist menu will contain only 1 TB and 2 TB as options. RAM picklist menu will contain 8 and 16 as options. Screen Size will contain 14 and 14 HD as options. The Fingerprint Reader picklist menu will contain Yes and No as options.

To create an Attribute Value Matrix

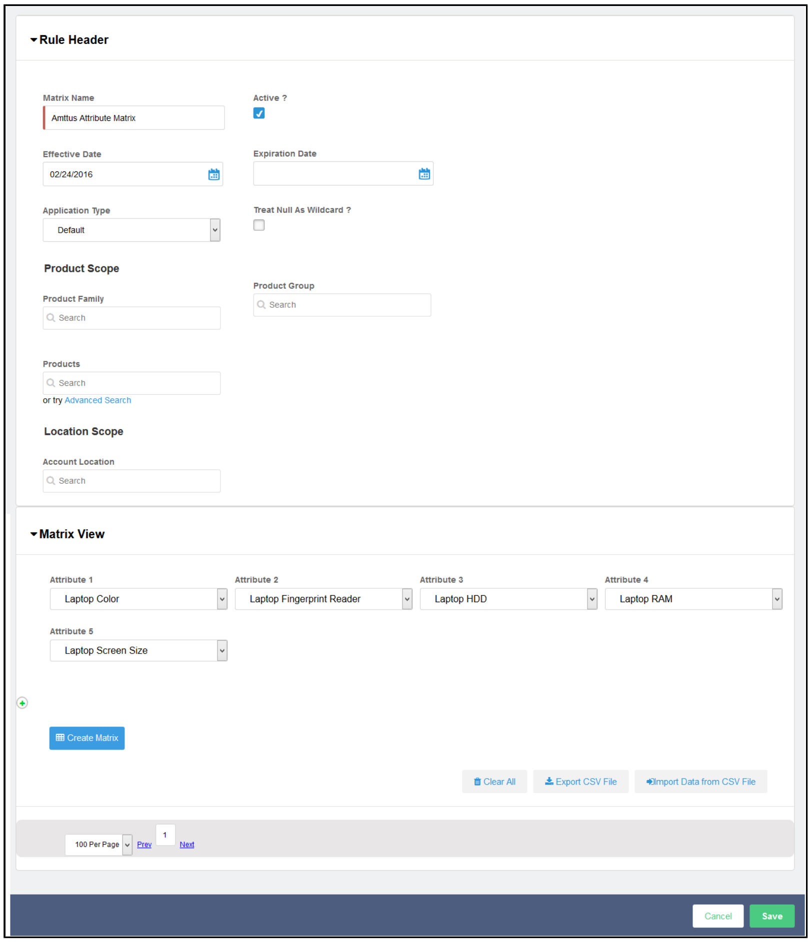
1. Click the **Attribute Value Matrices** tab.
2. Click **New**.
3. Provide requisite information in the Rule Header section for the following fields.

Field Name	Description

Name	
Rule Header	
Matrix Name	Enter Name of the Attribute Value Matrix.
Active	Flags whether the Attribute Value Matrix is active.
Effective Date	Select the date from which the Attribute Value Matrix is in effect.
Expiration Date	Select the date until which the attribute value matrix is valid.
Application Type	A picklist field, which contains *Default*, *Constraint* and *Force Set* values. *Default*: Selecting this lets you choose the picklist values which you have configured as default values in the Matrix View. The rest of the values are visible in the drop-down list, but you cannot select any of them on the Cart. *Constraint*: Selecting this value lets you choose only those picklist values which you have configured in the Matrix View. All the other values are disabled in the drop-down list on the cart. This also works with the Multi-select picklist. *Force Set:* Selecting this value lets you automatically set the picklist values which you have configured in the Matrix View (considering the last user selection). As soon as you change the attribute value for one attribute, the system sets the attribute values for other attributes immediately. If there is a Matrix entry with null values and **Treat Null As Wildcard** is set to false, the system resets the attribute values to null.
Treat Null As Wildcard	This field works in conjunction with **Application Type**. If you select this check box, all the picklist values (default or constrained) configured in the Matrix View are available for selection on the Cart. Deselecting this check box disables all the picklist values on the Cart.

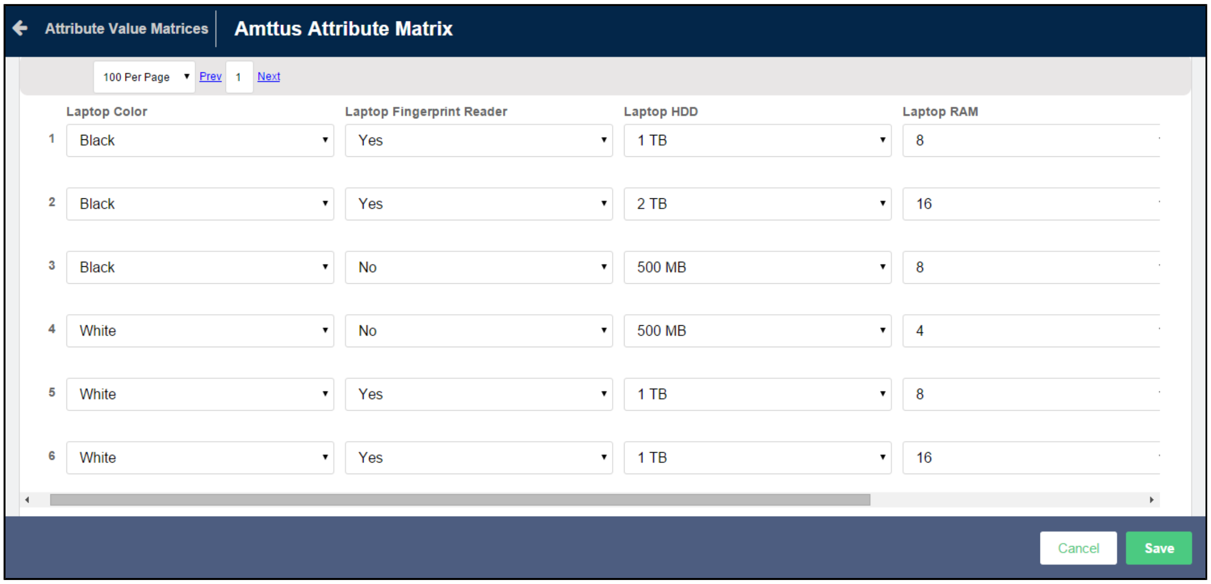
Field Name	Description
Product Family	A multi-select field. Click in the text box to view the available product families and select the appropriate ones. You can also enter a keyword and search for a Product Family.
Product Group	A multi-select field. Click in the text box to view the available product groups and select the appropriate ones. You can also enter a keyword and search for a product group. The product groups are filtered on the basis of product families that you select.
Products	A multi-select field. Click in the text box to view the available products and select the appropriate ones. You can also enter a keyword and search for a product. The products are filtered on the basis of product families and product groups that you select.
Advanced Search	Click Advanced Search link to view the details (Product Code, Product Name and Product Family) of the products available. You can select the appropriate products and click Select. Performing this step adds the products fot which this attribute value matrix is applicable.
Account Location	A multi-select field. Click in the text box to view the available account locations and select the appropriate ones. You can also enter a keyword and search for a location. The locations are filtered on the basis of product families, product groups, and products that you select.

4. Under the Matrix View section, type in the name of attribute that you want to associate.
5. Click **Clear All** to clear the attribute selection from the picklist menus.
   -or-
   Click **Export CSV file** to export the content of Matrix View as a CSV file.
   -or-
   Click **Import Data from CSV file** to import the data from an existing CSV file.
6. Click **Create Matrix**. This creates a matrix.



7. Scroll down to create multiple valid combinations from the picklist menus. You can select the appropriate attributes from the picklist menu.
   Note that these attribute picklists work in a bi-directional manner. For example, if you choose **Laptop Color** = Black, **Laptop HDD** is defaulted to *1TB.* Similarly, if you select **Laptop HDD**

as *1TB*, the system defaults **Laptop Color** to *Black*.



8. Click **Save**.

You must run attribute maintenance jobs after creating an attribute matrix. Even after you edit the criteria, you must run **Criteria Maintenance > Update Expression Fields** and then **Attribute Maintenance**.

To run attribute maintenance

1. Click the **Attribute Maintenance** tab.
   -or-
   Click and click **Attribute Maintenance**.
2. Click **Update Attributes**.

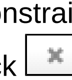
Creating Product Attribute Rules

Product Attribute Rules allow product configuration according to product attributes that are based on the actions you select. Each attribute value selection should ensure that the rest of the attributes get constrained to allow only valid selections to help narrow the result set. The attributes can also leverage expressions and calculations to derive values that help identify the product variant. This feature simplifies the configuration process, eliminates the need for component-level choices and provides guided selling at the product configuration level. You can also set multiple actions for the same product attribute rule.

The product attribute rule is applied to the criteria that you define in the Criteria section. You can define simple and advanced criteria. In simple criteria, you can select field, operator and value. In advanced criteria, you can set a condition, if it is true, then the actions in the Action section are triggered.

> **Note**
>
> When you upgrade to the November 2016 release, ensure that you change the value of **Product Attribute Rule Action** object **> Action** picklist from *Read Only* to *Disabled*. This is required so that the Product Attribute Rule functions correctly.

If you have set a default value for a field from Salesforce and you define a product attribute rule with an action of setting the default value, then the value mentioned in the Salesforce field is set. In short, you cannot override the default value that is set in the Salesforce field by using a product attribute rule.

> **Note**
>
> Attributes are constrained on the basis of selection of other attributes. The constrained values are not displayed in the drop-down menu. In order to reset the list, click  or select -None- from the picklist.

For example, Amttus, a leading laptop manufacturer, wants to drive the product selection of their users by using attributes. A laptop has five attributes - Color, HDD, RAM, Screen Size, and Fingerprint Reader. The following table describes the products and its attributes.

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB
RAM	8	16	4	8	8	16

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

Now, we want to set the default value of Fingerprint Reader as yes, if the user selects White from Color picklist menu and 500 MB from HDD picklist menu.

To create a product attribute rule

1. Click the **Product Attribute Rules** tab.
2. Click **New Product Attribute Rule**.
   You can also clone an existing product attribute rule using **Clone (With Actions)** button. This button clones the information in the **Actions** related list also.
   You must click **Edit Layout** on the detail screen and bring this button on your page layout.
3. Provide requisite information in the Rule Header section for the following fields.

Field Name	Description
Rule Header	
Product Attribute Rule Name	Enter the Name of Product Attribute Rule.
Active	Flags whether the product attribute rule is active.
Effective Date	Select the date when the product attribute rule is in effect. Leave this field blank if you want this product attribute rule to be always effective.
Expiration Date	Select the date when the product attribute rule will become invalid. Leave this field blank if you want this product attribute rule to be always effective.

Field Name	Description
Product Family	A multi-select field. Click in the text box to view the available product families and select the appropriate ones. You can also enter a keyword and search for a product family.
Product Group	A multi-select field. Click in the text box to view the available product groups and select the appropriate ones. You can also enter a keyword and search for a product group. The product groups are filtered on the basis of product families that you select.
Products	A multi-select field. Click in the text box to view the available products and select the appropriate ones. You can also enter a keyword and search for a product. The products are filtered on the basis of product families and product groups that you select.
Account Location	A multi-select field. Click in the text box to view the available account locations and select the appropriate ones. You can also enter a keyword and search for a location. The locations are filtered on the basis of product families, product groups, and products that you select.

4. Under the Criteria section, specify the formula to be evaluated. When this formula evaluates to true, the rule is executed.

Criteria	
Formula evaluates to true	Enter the formula to be evaluated.

5. Provide requisite information in the Action section for the following fields.

Field Name	Description
Action	

<table>
<tr><th>Field Name</th><th>Description</th></tr>
<tr><td>Target Attribute</td><td>Select the target attribute. Use type ahead to display a list of applicable fields or attributes.</td></tr>
<tr><td>Action Type</td><td>Select the action that you want to perform on the product attribute. Available options are:
Allow - To restrict the visibility of values on the target field.
Default - To add a default value on the target attribute.
Hidden - To hide the target attribute.
Disabled - To make the target attribute as a read-only field.
Required - To make the target attribute as a required field.
Reset - To auto-populate a default value on the target attribute if the field is left blank.
Click the relevant link to see the individual examples.</td></tr>
<tr><td>Value Expression</td><td>Enter the Value Expression. You can use the methods such as JOIN and DISCRETEINTERVAL.

DISCRETEINTERVAL(start, stop, interval): This method returns a discrete array of values from start to stop separated by interval.
Params:
start - the first element of the specified dimension in the array
stop - the last element of the specified dimension in the array
interval - the distance between two elements in the array
JOIN(array, separator): This method returns a concatenated text string of all the elements of an array, using the specified separator between each element.
Params:
array - an array that contains the elements to concatenate.</td></tr>
</table>

Field Name	Description
	• separator - the string to use as a delimiter between values. Separator is included in the returned string only if the value has more than one element. If no separator is specified, the default will be a comma (,). For example, if you want to limit the deductible amount (an attribute) for California state from 200 to 1000, with an interval of 100, use the expression as: *JOIN(DISCRETEINTERVAL(200,1000,100),';')*

- Click **Save**.

> Once you save your record, u pdate the **Target Attribute** with the prefix *Apttus_Config2__ProductAttributeValue__c*. For example, if your **Target Attribute** is *ProductAttributeValueId1__r.Ext1Multiselect__c*, after the update, your **Target Attribute** should be *Apttus_Config2__ProductAttributeValue__c. ProductAttributeValueId1__r.Ext1Multiselect__c*. This up-gradation is required to work with PAV Extension Objects.

Conditionally Allow Values for a picklist or multi-select picklist attribute

Attribute-Based Configuration allows you to create the dependency for the target attribute. You can configure if you want to allow the attribute values based on a condition. The constrained value should be derived from a value expression, for example, can be pulled from a product field. This helps in providing only a relevant set of attributes for the user to make selections, thus providing a guided experience for configuration.

The following table displays the data types on which you can perform this action and the value of Value Expression required.

Action	Data Types supported	Value Expression
Allow	Picklist Multi-picklist	The values to be allowed. 'Red;Blue;Black;Orange' For example, '2014;2012;2015'

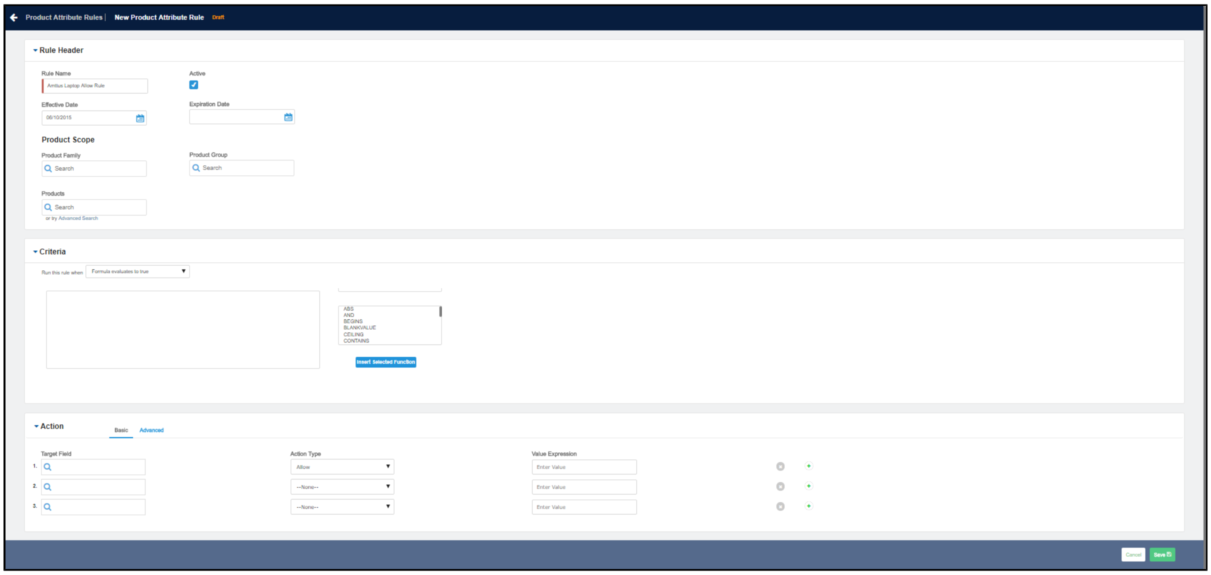
For example, Amttus, a leading laptop manufacturer, wants to use attributes to drive the product selection of their users. A laptop has five attributes - Color, HDD, RAM, Screen Size, and Fingerprint Reader. The following table describes the products and its attributes.

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB
RAM	8	16	4	8	8	16
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

The following table describes scenarios where you can use this feature.

Scenario	Target Attribute	Value Expression
When the user selects **White** color, you want to allow **Yes** as a Fingerprint reader attribute.	Laptop Fingerprint Reader	'Yes'

The following image displays the Product Attribute Rule for **Allow** action.



> **Note**
>
> If you are using IE browser version 11 and the classic UI, the system does not restrict the picklist values that you have configured using the above *Allow* action.

Conditionally Set Default Values for the Attributes

Attribute-Based Configuration allows you to create the dependency for the target attribute if that attribute should conditionally have a default value set. This feature helps provide starting point configurations for the product and reduce the number of clicks for the user.

> **Note**
>
> If you have set a default value of an attribute using numeric expression and you set the default value of the same attribute using Product Attribute Rule, then the attribute will populate the default value that is set by numeric expression. That is, the value mentioned in numeric expression has priority.
>
> Similarly, if you have set a default value of an attribute from Salesforce field and you set the default value of the same attribute using the Product Attribute Rule, then the attribute will populate the default value that is set using the Salesforce field. That is, the value mentioned in Salesforce field has priority.

The default values are set only when the attributes have no value. For example, you have selected *None* as picklist value. If you have set a value manually in your attribute, then the default value does not override it.

> **Note**
>
> When you are setting the default value for the target field, the following points must be kept in mind.
>
> - The default value for number Data Type must be written as mentioned in the field value.
> - The default value for text and picklist Data Type must be written between single quotes.
> - The default value for multi-picklist Data Type must be written between single quotes and must be separated by a semicolon without any space.

The following table displays the datatypes on which you can perform this action and the value of Value Expression required.

Action	Datatypes supported	Value Expression	Example
Default	Number Text Picklist Multi-picklist	The default value. For example, 'RED'	Number: 79 Text: 'Red' Picklist: 'Blue' Multi-picklist: 'Red';'Black'

For example, Amttus, a leading laptop manufacturer, wants to use the attributes to drive the product selection of their users. A laptop has five attributes: Color, HDD, RAM, Screen Size, and, Fingerprint Reader. The following table describes the products and its attributes.

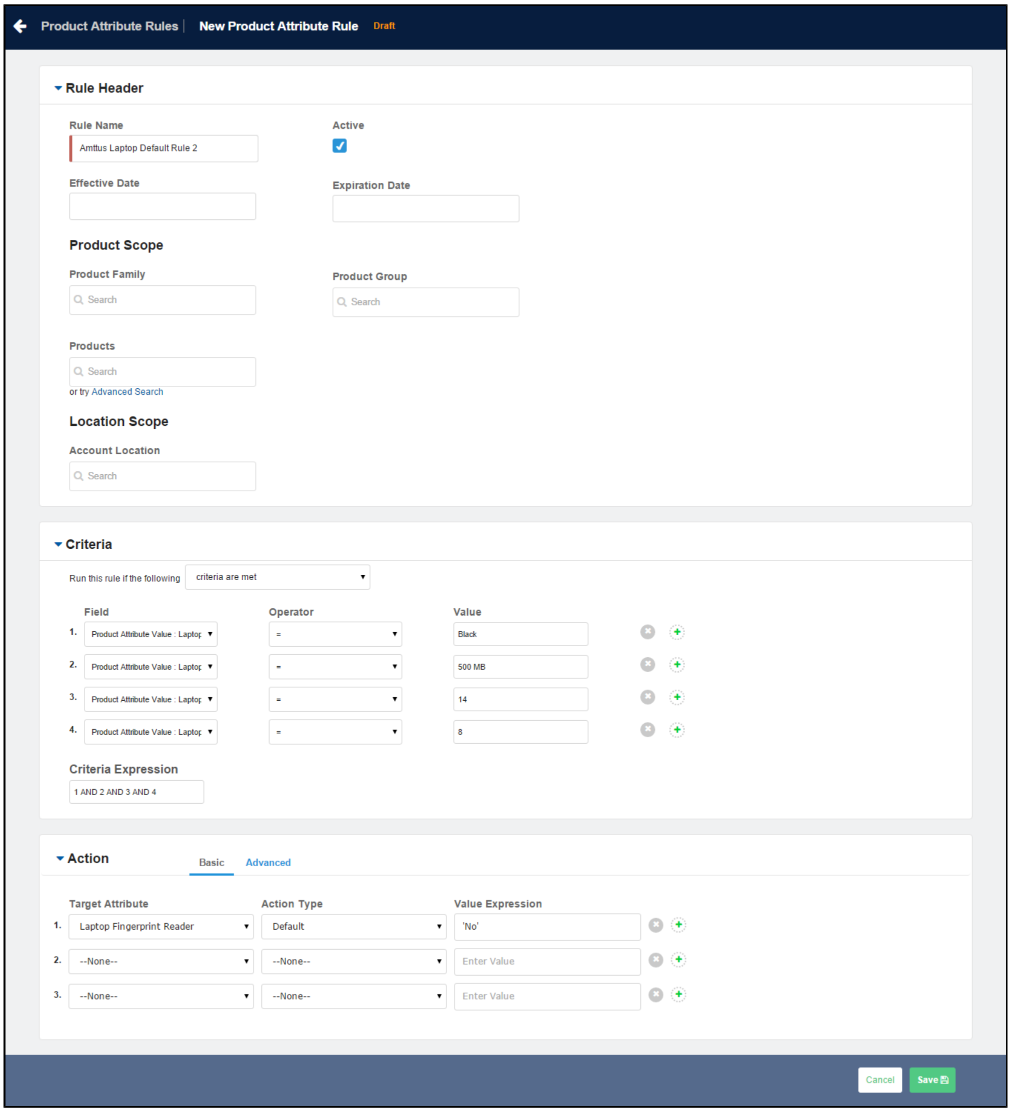
Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
RAM	8	16	4	8	8	16
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

The following table describes scenarios where you can use this feature.

Scenario	Target Attribute	Value Expression
When the user selects 2 TB from Laptop HDD picklist, Laptop RAM must set 16 as the default value.	Laptop RAM	16
When the user selects Black from Color, 1 TB from HDD, 8 from RAM, and 14 from Screen Size, Laptop Fingerprint Reader must set No as the default value.	Laptop Fingerprint Reader	'No'

The image below illustrates the Product Attribute Rule of **Default** action.



To disallow the value to be changed by a user, create a read-only attribute or make it conditionally read-only. For more information on conditionally making an attribute read-only, refer to Conditionally Make an Attribute as Read-Only.

Conditionally Hide Attributes

Attribute-Based Configuration allows you to create the dependency for the target attribute to either make the attribute conditionally visible or hide the attribute based on other selections. This helps to provide only relevant sets of attributes for the user to make selections, thus providing a guided experience for configuration.

The following table displays the data types on which you can perform this action and the value of Value Expression required.

Action	Datatypes supported	Value Expression
Make Hidden	Number Text Picklist Multi-picklist	

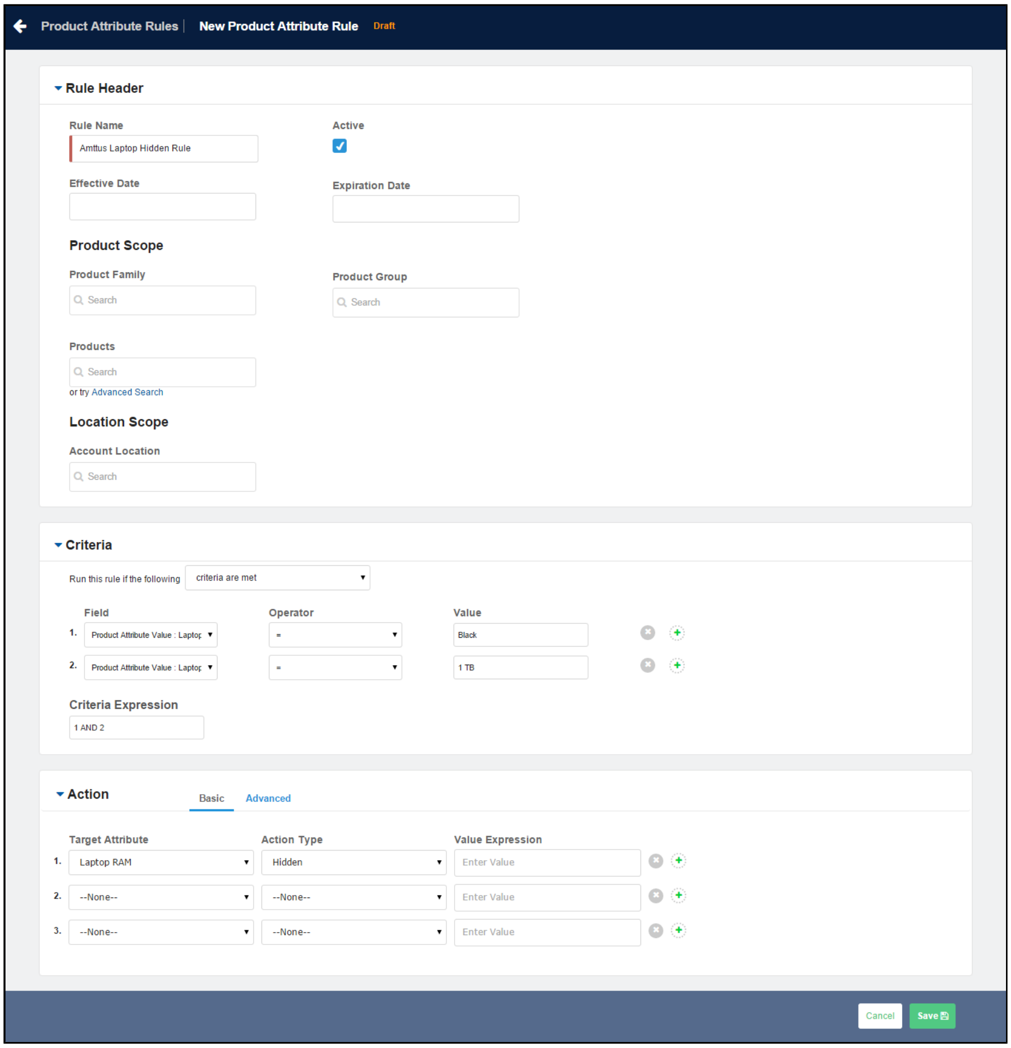
For example, Amttus, a leading laptop manufacturer, wants to use the attributes to drive the product selection of their users. A laptop has five attributes: Color, HDD, RAM, Screen Size, and Fingerprint Reader. The following table describes the products and attributes.

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB
RAM	8	16	4	8	8	16
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

The following table describes scenarios where you can use this feature.

Scenario	Target Attribute	Value Expression
When the user selects 4 from RAM, you want to hide the Screen Size attribute.	Screen Size	
When the user selects Black from Color and 1 TB from HDD, you want to hide the RAM attribute.	RAM	

The following image illustrates the Product Attribute Rule for **Hidden** action.



Conditionally Set an Attribute as Disabled

Attribute-Based Configuration allows you to create the dependency for the target attribute if that attribute should conditionally be marked as *Disabled*. This feature prevents the the user from changing a value of an attribute.

The following table displays the data types on which you can perform this action and the value of Value Expression required.

Action	Data Types supported	Value Expression
Disabled	Number Text Picklist Multi-picklist	

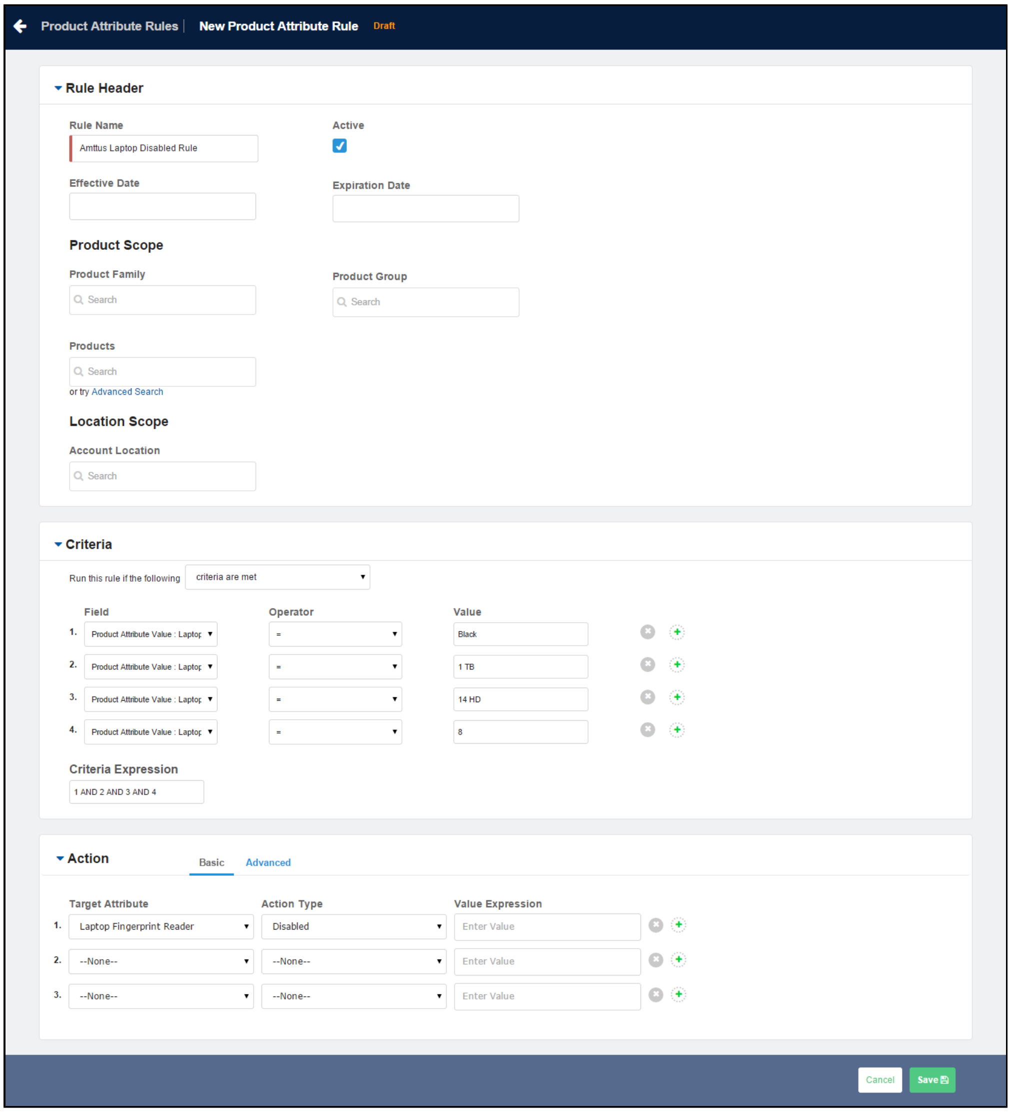
For example, Amttus, a leading laptop manufacturer, wants to use attributes to drive the product selection of their users. A laptop has five attributes: Color, HDD, RAM, Screen Size, and Fingerprint Reader. The following table describes the products and its attributes.

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB
RAM	8	16	4	8	8	16
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

The following table describes scenarios where you can use this feature.

Scenario	Target Attribute	Value Expression
When the user selects Black from Color, 1 TB from HDD, 8 from RAM, and 14 HD from Screen Size, the Fingerprint Reader must be disabled.	Laptop Fingerprint Reader	

The following image illustrates the Product Attribute Rule for **Disabled** action.



Conditionally Set an Attribute as Required

Attribute-Based Configuration allows you to create the dependency for the target attribute if an attribute must be made conditionally mandatory while selection. This helps to prevent the user from missing any important attribute that must be entered.

The system will auto-select the remaining value if only one value is remaining in a **Required** attribute after the constraints are fired via a matrix or a product attribute rule.

The following table displays the data types on which you can perform this action and the value of Value Expression required.

Action	Datatypes supported	Value Expression
Make Required	Number Text Picklist Multi-picklist	

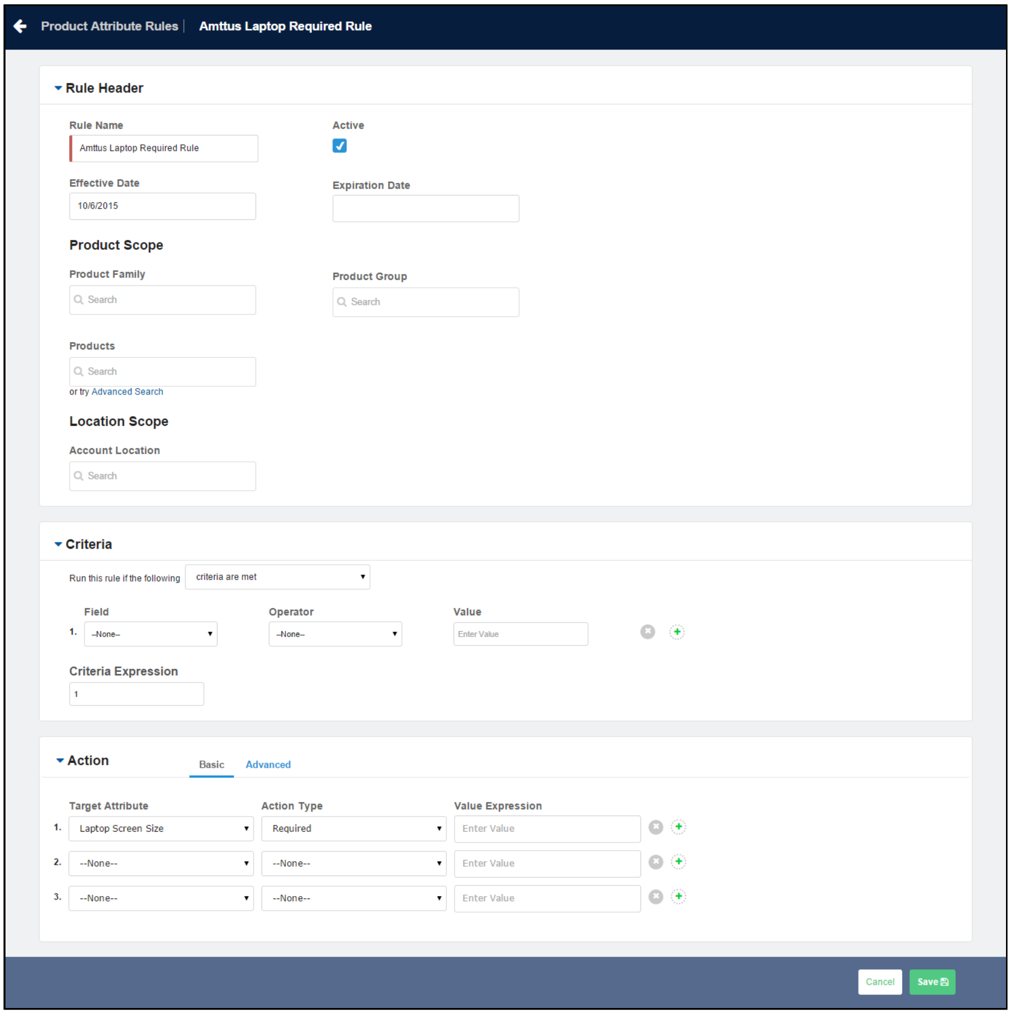
For example, Amttus, a leading laptop manufacturer, wants to use the attributes to drive the product selection of their users. A laptop has five attributes: Color, HDD, RAM, Screen Size, and Fingerprint Reader. The following table describes the products and its attributes.

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB
RAM	8	16	4	8	8	16
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

The following table describes scenarios where you can use this feature.

Scenario	Target Attribute	Value Expression
You want to ensure that the user selects Screen Size.	Laptop Screen Size	

The following image illustrates the Product Attribute Rule for **Required** action.



Conditionally Reset an attribute

Attribute-Based Configuration allows you to create the dependency for the target attribute to conditionally reset an attribute based on other selections. This helps in ensuring a default value to be added to the target attribute if the user leaves the attribute blank or null. If the user does not select any value in the attribute, the default value be selected by default.

The following table displays the datatypes on which you can perform this action and the value of Value Expression required.

Action	Datatypes supported	Value Expression
Reset	Number	The default value.

Action	Datatypes supported	Value Expression
	Text Picklist Multi-picklist	For example, 'White' .

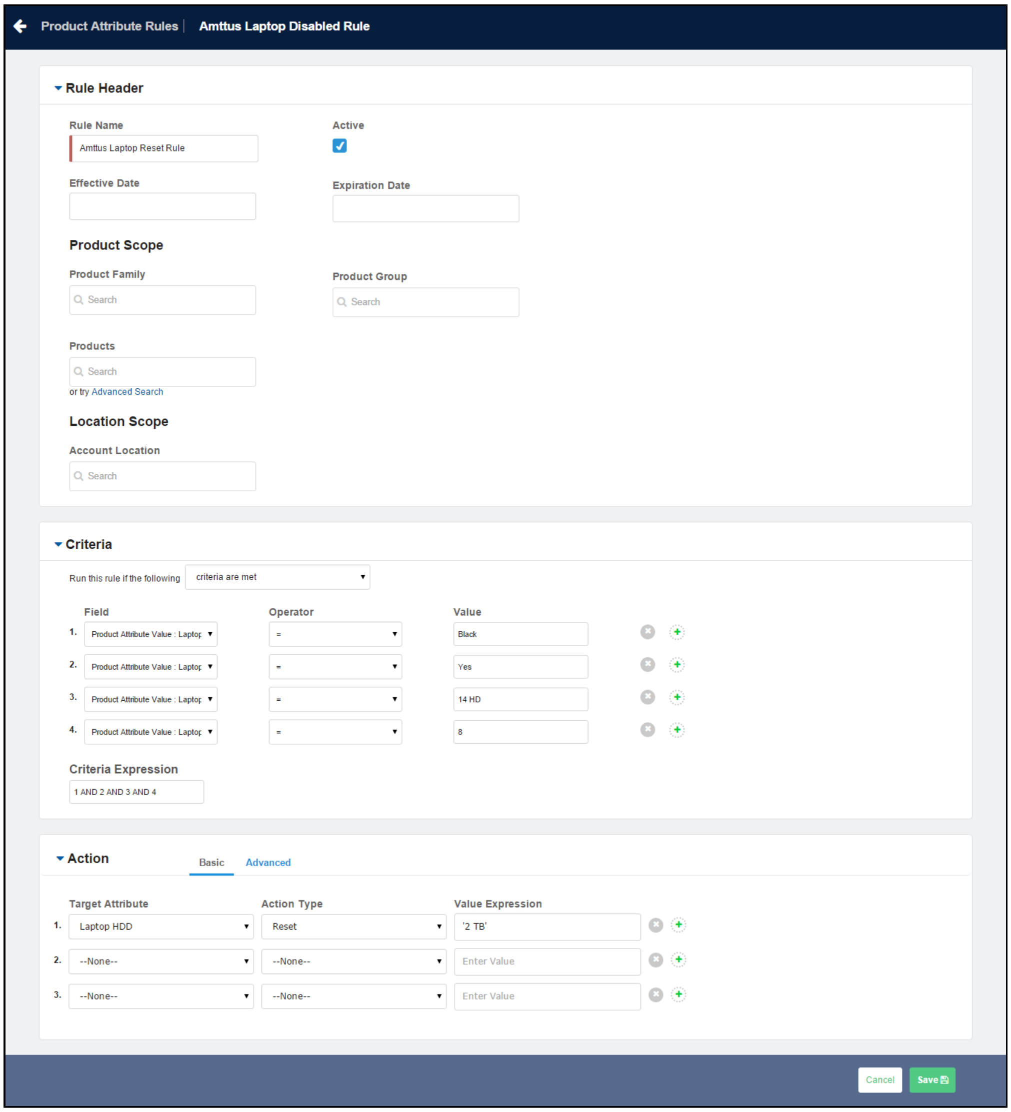
For example, Amttus, a leading laptop manufacturer, wants to use the attributes to drive the product selection of their users. A laptop has five attributes: Color, HDD, RAM, Screen Size, and Fingerprint Reader. The following table describes the products and its attributes.

Product	A-560	A-560m	A-440	A-440m	A-320	A-320m
Color	Black	Black	White	White	Black	White
HDD	1 TB	2 TB	500 MB	500 MB	1 TB	1 TB
RAM	8	16	4	8	8	16
Screen Size	14 HD	14 HD	15	15 HD	14	14 HD
Fingerprint Reader	Yes	Yes	Yes	Yes	No	Yes

The following table describes scenarios where you can use this feature.

Sceanrio	Target Attribute	Value Expression
When the user selects Black from Color, 16 from RAM, 14 HD from Screen Size, and Yes from Fingerprint Reader, you want to reset the value of HDD to 2 TB.	Laptop HDD	'2 TB'

The following image illustrates the Product Attribute Rule for **Reset** action.



Configuring Product Comparison on the Catalog Page

You can enable an end user to compare features of multiple products on the catalog page.

In order to configure product comparison on the catalog page, you must set up feature sets with one or more features. You must associate these feature sets to one or more products and then define feature values for each product. The configuration allows you to enable or disable a product feature to be displayed during product comparison.

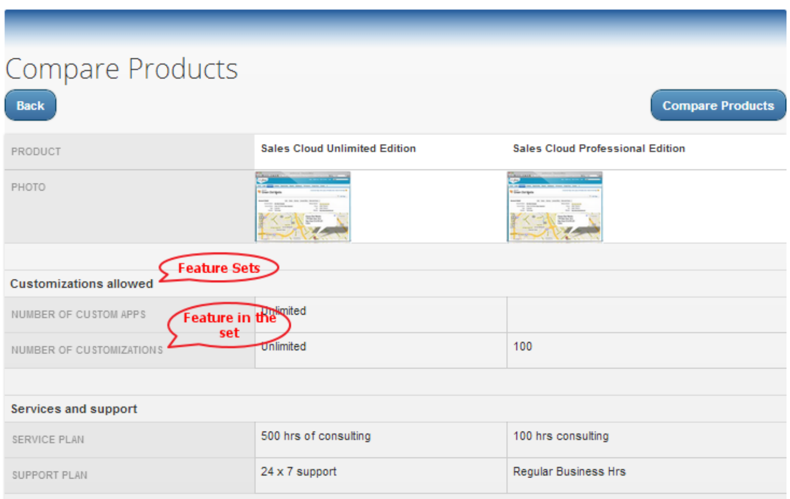
- Setting Up Feature Sets
- Associating a Feature Set to a Product
- Define Feature Values for a Product

If you do not want to use the Compare Products Feature, you can hide the button from the Catalog page.

Setting Up Feature Sets

Feature sets are a collection of features that can be associated to a product.

You must first define a feature set and then define features for each set.



To create feature sets

1. Click ✚ and click **CPQ Console**.
2. From the Product Management section, click **Add Feature Set**.
3. Type a mandatory **Feature Set Name**, and type a Description.
4. In the Feature Information section, type a **Feature Name** and a Description.
5. To add more than one feature, click **New**. A new row is added. The sequence number is auto-populated. You can rearrange the sequence by clicking and dragging the row.

6. Click **Save**.

A feature set is set up with features.

Associating a Feature Set to a Product

You must associate a feature set to a product for product comparison on the catalog page.

Using Product Console, you must associate a feature set to a standalone and a bundle product.

To associate a feature set to a product

You must have an existing product.

1. From the CPQ Console, click **Manage Products** and select an existing product to associate a feature set.
2. Click the **Product Console** button and scroll down to the Additional Data section, and click **Features**.
3. Click **Manage Associations** and from the Available Groups list, select a feature set for association and click **Add**.

The feature set is associated to a product.

Define Values for the Product Features

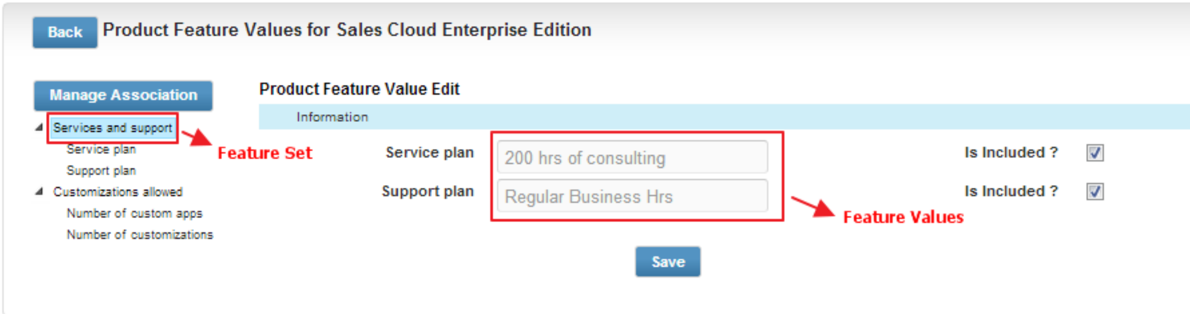
Define values for the features you just created.

You must define the feature values for a given product.

To define feature values for a product

You must have an existing feature set and must be associated to a product.

1. On the Product Feature Values page, select the feature set listed in the left pane and type values next to the each feature.



2. To display the feature on the product comparison page, select **Is Included** and click **Save**.

Features Values that you just defined are saved.

Hiding Compare Products Button on the Catalog Page

There is a custom setting that enables you to show or hide the *Compare Products* button and the product compare check box in the Catalog page.

After applying the new setting, you will no longer see the *Compare Products* button and the item level product compare check boxes.

To hide the compare products button on the catalog page

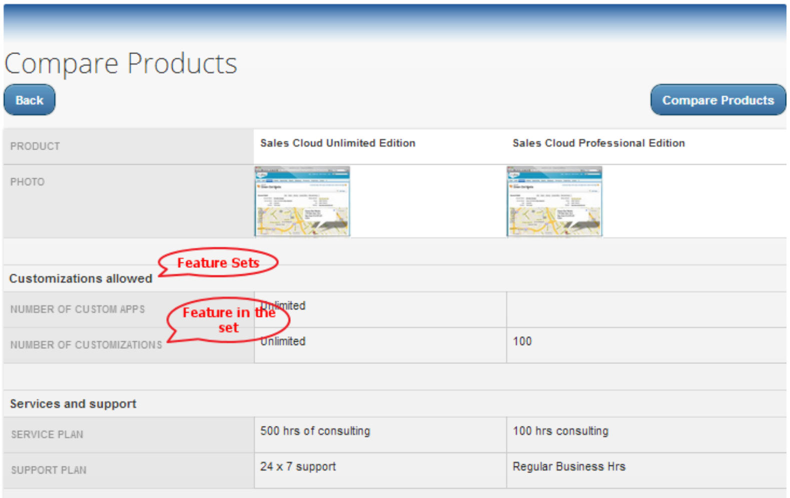
1. Click ✚ and click **Config Settings**.
2. Click **Catalog Page Settings** and from the Product List Settings section, select Hide Compare Products to hide the compare products feature on the catalog page.
3. Click **Save**.

The compare products feature is not displayed on the catalog page.

Setting Up Feature Sets

Feature sets are a collection of features that can be associated to a product.

You must first define a feature set and then define features for each set.



To create feature sets

1. Click and click **CPQ Console**.
2. From the Product Management section, click **Add Feature Set**.
3. Type a mandatory **Feature Set Name**, and type a Description.
4. In the Feature Information section, type a **Feature Name** and a Description.
5. To add more than one feature, click **New**. A new row is added. The sequence number is auto-populated. You can rearrange the sequence by clicking and dragging the row.
6. Click **Save**.

A feature set is set up with features.

You must associate a feature set to a product for product comparison on the catalog page.

Associating a Feature Set to a Product

Using Product Console, you must associate a feature set to a standalone and a bundle product.

To associate a feature set to a product

You must have an existing product.

1. From the CPQ Console, click **Manage Products** and select an existing product to associate a feature set.

2. Click the **Product Console** button and scroll down to the Additional Data section, and click **Features**.

3. Click **Manage Associations** and from the Available Groups list, select a feature set for association and click **Add**.

The feature set is associated to a product.

Define values for the features you just created.

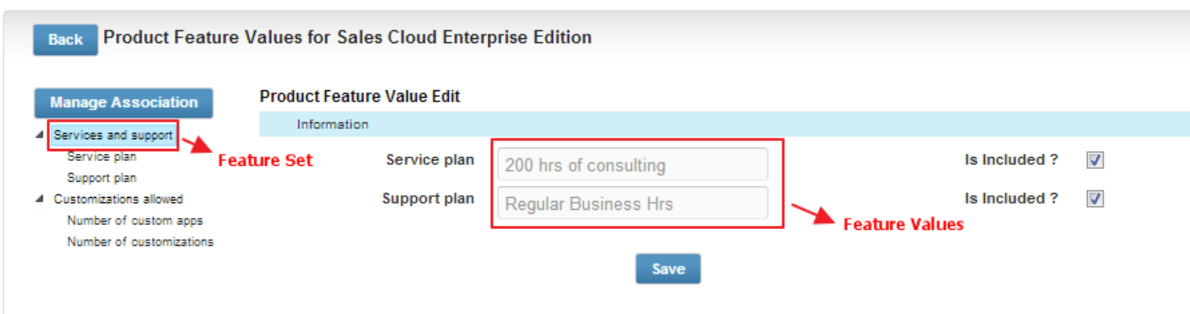
Define Feature Values for a Product

You must define the feature values for a given product.

To define feature values for a product

You must have an existing feature set and must be associated to a product.

1. On the Product Feature Values page, select the feature set listed in the left pane and type values next to the each feature.



2. To display the feature on the product comparison page, select **Is Included** and click **Save**.

Features Values that you just defined are saved.

Hiding Compare Products Button on the Catalog Page

There is a custom setting that enables you to show or hide the *Compare Products* button and the product compare check box in the Catalog page.

After applying the new setting, you will no longer see the *Compare Products* button and the item level product compare check boxes.

To hide the compare products button on the catalog page

1. Click and click **Config Settings**.
2. Click **Catalog Page Settings** and from the Product List Settings section, select **Hide Compare Products** to hide the compare products feature on the catalog page.
3. Click **Save**.

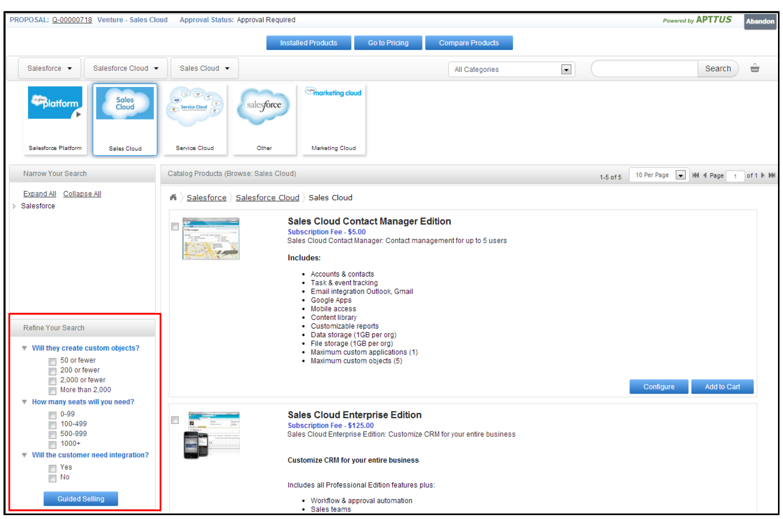
The compare products feature is not displayed on the catalog page.

Refining Your Search

When you are selecting items from the Product Catalog, *Refine Your Search* enables you to narrow the list of products to those that match your search filter questions.

Search filters are associated with categories, so that the *Refine Your Search* questions may change when you select different product categories. The questions are also found on the product record, which is how the Product list gets refined. When you select an answer to a question and that answer matches the answer on the product record, that product is included in the filtered Product list.

You can select multiple values from the picklist for refining your search. The search is restrictive and helps narrow down the result to show only the relevant data. The Product field value should contain the filter value. The search filter provides all possible values. For example, if you select Asia and Europe from Regions drop-down menu, then products available in both these regions are shortlisted.



> **Note**
>
> For a product to be displayed it must match all of the selected *Refine Your Search* criteria.

To configure your system to use Refine Your Search you must complete the following:

Configure the custom fields	Two custom fields need to be created. The field for Search Attribute Values is used to provide the search question and answers, which will be associated with a category. The field for Products is used to have the same search question and answer values available on the product record.
Configure search filter fields	This makes the custom field available from the *Search Filter Fields* list in the Hierarchy Manager . This must be completed so that the custom field can be associated with a specific category.
Configure refine search options	This task associates the custom field with a specific category and adds it to the Refine Your Search window in the Product Selection page.
Select a search field value for a product	So that using *Refine Your Search* provides useful and relevant filtering, you should select a value for each search field on a product record.

When you are selecting items from the Catalog page, *Refine Your Search* enables you to narrow the list of products to those that match your search filters. The system can dynamically display only those filter values that are based on the products associated with the selected category.

Configure the custom fields

To configure the custom fields

1. Go to App Setup > Create > Objects and select Search Attribute Value.
2. Go to Custom Fields & Relationships and click New.

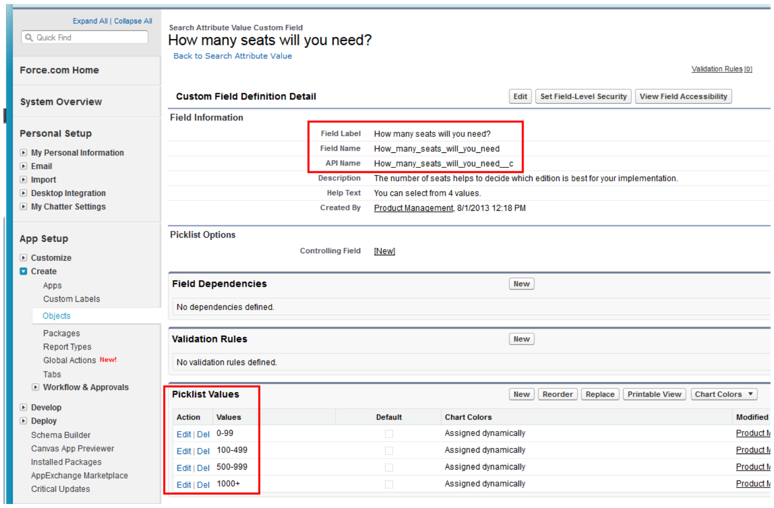
3. Select Picklist data type and click Next.
4. Enter the details:

Option	Description
Field Label	This is the question. This is the text that will be displayed in the *Refine Your Search* box when you are selecting your products. It is also displayed in the product's *Product Elements* section.
List of values	These are the answers. From the Product Catalog these are the options you can choose from to refine your search on the *Products* page. From the product record you can select one of these values for the product.
Field Name	This is the basis for the Salesforce API name, which is referenced in *Search Filter Fields*. The API name takes the field name and appends__cto it.
Description & Help Text	Standard Salesforce description and help text fields.

   Click Next.

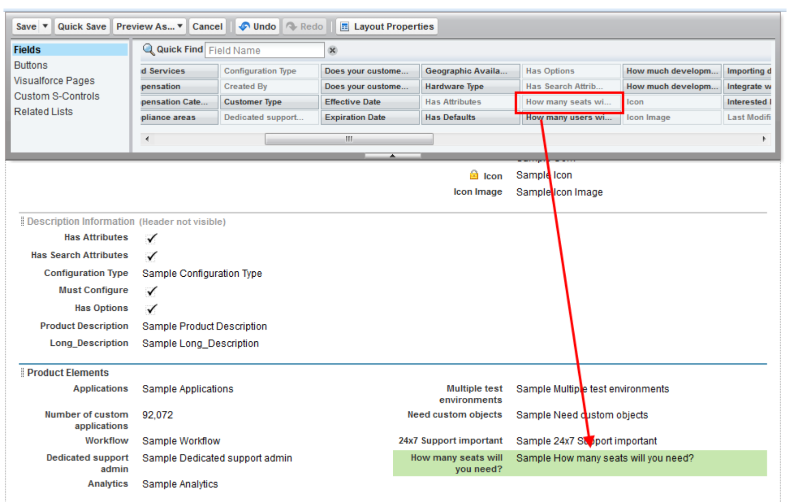
5. Select which profiles will have access to the field and click Next.
6. Ensure the field is added to available layout and click Save.

The new custom field is created and can now be associated with the Category Hierarchy search fields.



You must now complete the same task again, creating the same field label, values, and field name, but for Products. Go to App Setup > Customize > Products > Fields and then begin from Step 2.

If you want to change the default placement of the fields, you can reorganize them, by editing the page layout. To help group them together, typically the products custom field would be placed in a *Product Elements* section for the *Products* page layout.



Configure search filter fields

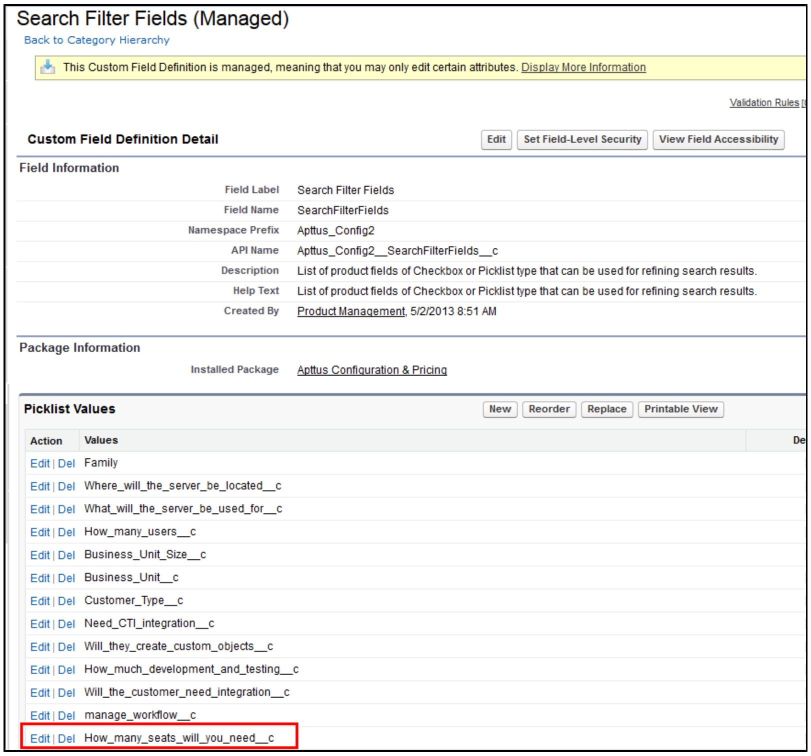
To configure search filter fields

The custom field must have already been created.

1. Go to App Setup > Create > Objects and select Category Hierarchy.

2. Scroll to Custom Fields & Relationships and select Search Filter Fields.
3. Scroll to Picklist Values and click New.
4. Enter the *API Name* for the custom field created in the *Search Attribute Value* custom object and click Save.

That field is added to the list and is now available in the Hierarchy Manager.



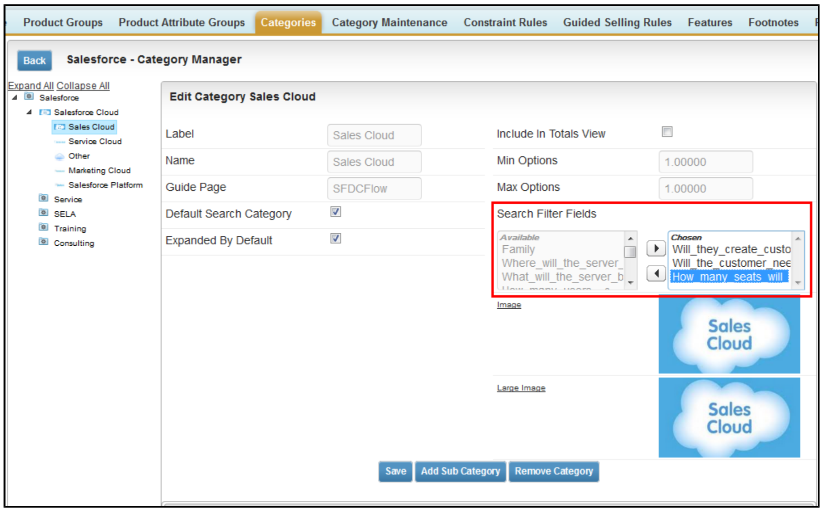
You can go to the Hierarchy Manager and add that field as a *Refine Your Search* question for specific categories.

Configure refine search options

To configure refine search options

You must have configured Search Attribute Value custom fields and Search Filter fields.

1. Go to CPQ Console > Manage Categories and select the category you want to have *Refine Your Search* questions for.
2. In the Category Detail section, click Hierarchy Manager.
3. From the Search Filter Fields list, select the refine search filters you want to use and click .



4. Click **Save**.

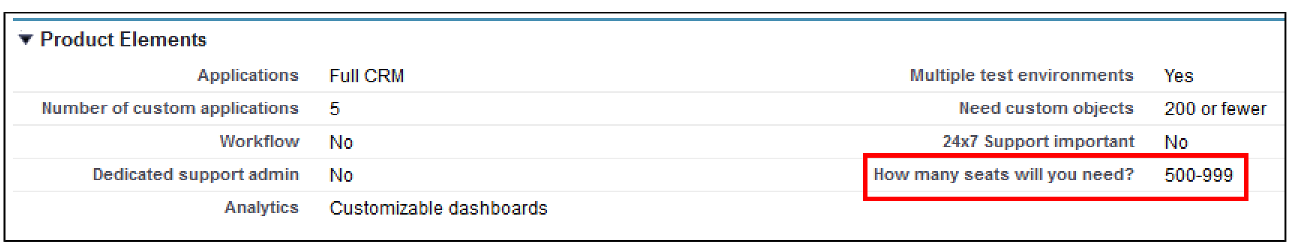
The next time someone goes to select products for that category, the selected search filter fields will be displayed.When desired, you can return to the Hierarchy Manager to remove or add additional search filters.

To select a search field value for a product

The custom search field must be available on the page layout used to display the products.

1. Select a product.
2. Click on the search field question and select the appropriate value for the product.
3. Click **Save.**

The value is displayed in the product record.



When that value is selected as a *Refine Your Search* answer, the product will be displayed in the refreshed Product list.

Refine Your Search With Dynamic Filter Values

When you are selecting items from the Catalog page, *Refine Your Search* enables you to narrow the list of products to those that match your search filters. The system now dynamically displays only those filter values that are based on the products associated with the selected category.

After you upgrade CPQ, you must run the Product Filter Maintenance task that collects all product field values related to a category. The product filter maintenance should be run when filter field association to the category changes or when the filter field values change in the product record. Also when new products are added or products are removed from a category you need to run the maintenance.

To run the Product Filter Maintenance task

You must have CPQ Summer13 SP5.5 installed.

1. Click ➕ and click **Product Filter Maintenance**.
2. Click **Update Product Filters**.

The batch job is completed collecting all the product field values related to a category.

Configuring Product Footnotes

Product Footnote object and two other objects: Proposal Footnote and Agreement Footnote enable you to dynamically include static *footnote* content in your quote/proposal documents, based on the products that are included in the quote as Proposal Line Items.

When a quote is finalized, the system creates corresponding Proposal Footnotes records. Merge Fields for this related object must be used in the proposal type templates. Similarly, for the agreement documents and agreement type templates merge fields from the Agreement Footnotes object must be used for the system to bring footnotes for the included Agreement Line Items.

Currently, the system supports multiple Footnotes for the same product; each being a text for up to 4000 characters on separate lines. Only unique footnote records are brought by the system to the context object level on finalization. Footnotes for the line items of the type: Product/Service, i.e., standalone products or bundles, are supported.

To create product footnotes

1. Select a product and from the Product Detail section, click **Product Console**.
2. Scroll to Additional Data and click **Terms** to display the New Footnote page.

3. Type a **Footnote Name** and type the text of the footnote in **Body**.
4. Type a mandatory **Sequence** and click **Save New Footnote**.

The footnote record is saved and associated to with the product and automatically added to the Footnotes related list.

You can go to the New tab and add another footnote.

You can go the Associate tab and search for an existing footnote to add to this product's Footnotes related list. You can associate only one footnote per product, but a footnote can be associated to more than one product.

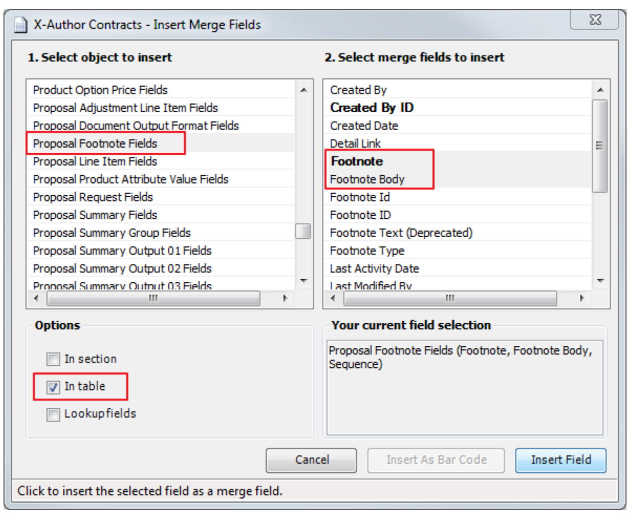
From the Edit tab you can make changes to the footnote or click Remove and choose to remove it from this product or any other products it has been associated with.

The newly created footnote can also be associated with other products.

To create a proposal type template

You must have existing product footnotes.

1. Login to X-Author Contracts, select the **X-Author Templates** tab and check out an existing proposal template (or create a new proposal type template).
2. Place the cursor in the template where you want the footnotes and click **Insert Merge Fields** .
3. From the *1. Select object to insert* pane, locate and select **Proposal Footnote Fields**.



4. From the *2. Select merge fields to insert* pane, select the fields that you want to insert.
5. From the *Options* section, select **In Table** and click **Insert Field**.

The merge fields are entered into the document template.

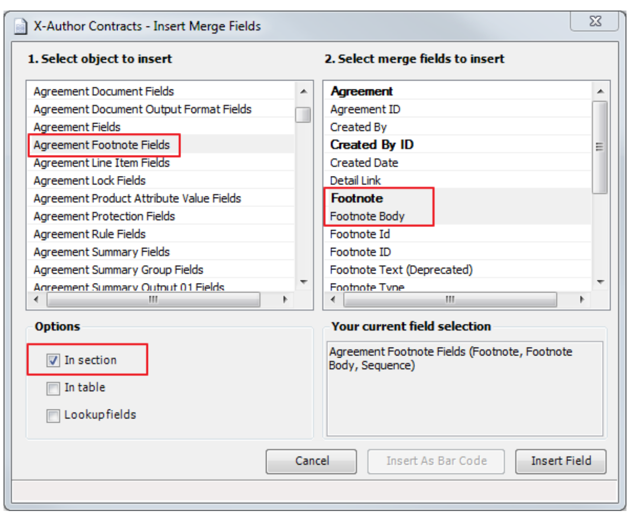
Proposal Footnotes in TABLE

Sequence	Footnote	Footnote Body
{ MERGEFIELD Apttus_QPConfig__ProposalFootnote_start * MERGEFORMAT }{ MERGEFIELD ProposalFootnote_Sequence * MERGEFORMAT }	{ MERGEFIELD ProposalFootnote_FootnoteId * MERGEFORMAT }	{ MERGEFIELD ProposalFootnote_FootnoteBody * MERGEFORMAT }{ MERGEFIELD Apttus_QPConfig__ProposalFootnote_end * MERGEFORMAT }

To create an agreement type template

You must have existing product footnotes.

1. Login to X-Author Contracts, select the **X-Author Templates** tab and check out an existing proposal template (or create a new proposal type template).
2. Place the cursor in the template where you want the footnotes and click **Insert Merge Fields** .
3. From the *1. Select object to insert* pane, locate and select **Agreement Footnote Fields**.



4. From the *2. Select merge fields to insert* pane, select the fields that you want to insert.
5. From the *Options* section, select **In Section** and click **Insert Field**.

The merge fields are entered into the agreement template.

Agreement Footnotes in Section

{ MERGEFIELD Apttus_CMConfig__AgreementFootnote_section_start * MERGEFORMAT }

{ MERGEFIELD AgreementFootnote_FootnoteId * MERGEFORMAT }

{ MERGEFIELD AgreementFootnote_FootnoteBody * MERGEFORMAT }

{ MERGEFIELD AgreementFootnote_Sequence * MERGEFORMAT }

{ MERGEFIELD Apttus_CMConfig__AgreementFootnote_section_end * MERGEFORMAT }

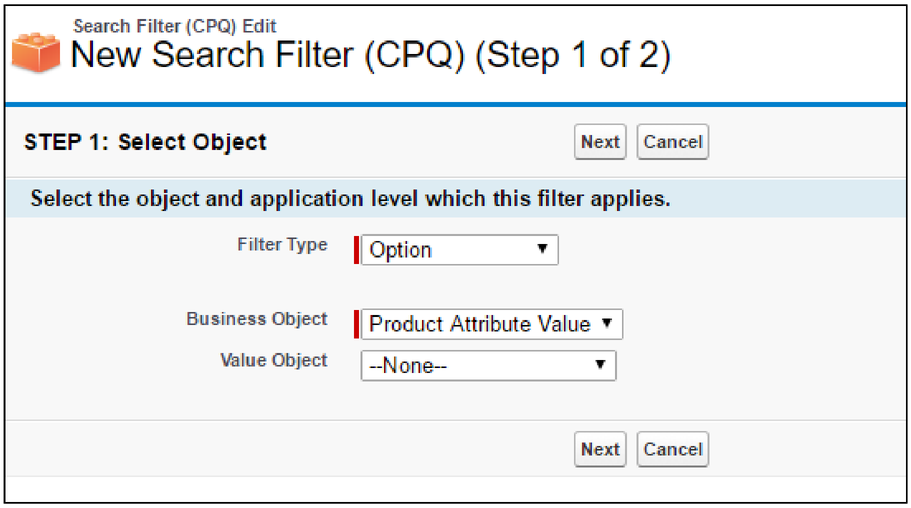
Configuring Conditional Option Inclusion

This feature allows you to perform auto-inclusion of an option based on search filter. Using this feature, you can efficiently auto-include options those are based on attributes.

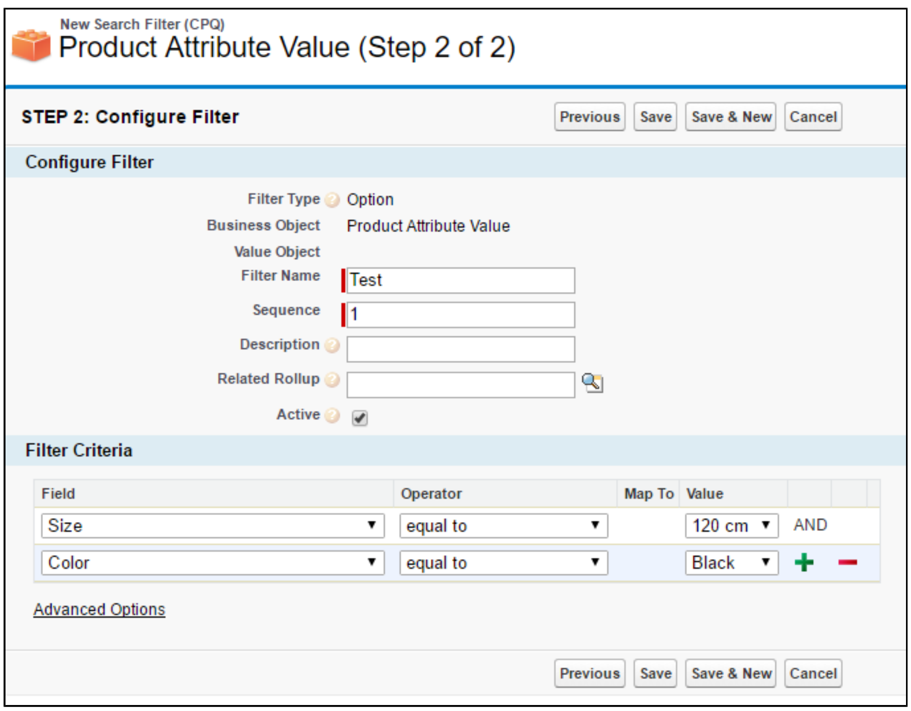
Use-Case: Option should be auto-included when an attribute is changed during the product configuration. As an alternative to using inclusion rules with criteria, as an Admin, you can configure options using **Search Filter (CPQ)** as a means to include options. This is only applicable for product attributes.

Setting up auto-inclusion of option using Search Filter (CPQ)

1. Navigate to **All Tabs > Search Filter (CPQ)**. Click **New**.
2. In **STEP 1: Select Object**, from **Filter Type**, select *Option*. From **Business Object**, select *Product Attribute Value*. Leave **Value Object** as *None.* Click **Next**.

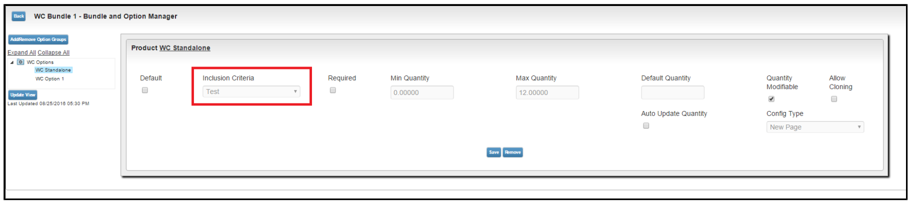


3. In **STEP 2: Configure Filter**, specify desired **Filter Name** and **Sequence** value. Ensure that **Active** checkbox is selected. From the **Filter Criteria** section, select the desired attribute from the **Field** and its evaluation criteria. You can use the + at the end of the line to create additional conditions. By default, they are filtered using AND condition. You can modify using the **Advanced Options**. Once you have configured the filter, click **Save**.



4. Run the **Product Filter Maintenance** batch job. To run this batch job, navigate to **Apttus Batch Updates > Product Filter Maintenance**, and click **Update Product Filters**.

5. Navigate to your bundle product's **Product Console** and click **Manage Bundles/Options**.
6. Select the desired option from the navigation panel, which you want to be auto-included based on the search filter that you just created. From the Inclusion Criteria drop down, select the name of the search filter. Click **Save**.



The option will be auto-included based on the attribute without using the constraint rule.

Configuring Numeric Expressions

Numeric expression is the extended custom implementation of SFDC formula builder, which is fine tuned by Apttus. Numeric expression follows the syntax of SFDC formula and it provides all logical, text, date & time, math functions along with standard arithmetic operations. Note that numeric expression works for number fields.

Where you can use numeric expressions?

- To set the attribute value
- To set the minimum, maximum, and total quantity at the option group level
- To set the minimum, maximum, and default quantity at the option group level
- To define numeric rollups at product, product group, and cart level
- To set line item fields of auto-included products through constraint rules
- To set the criteria of product attribute rule (PAR)

Before you start...

The Apttus Expression builder feature is disabled by default. To enable this feature, click **Custom Settings > Config System Properties > System Properties > Enable Field Expressions**.

Consider a requirement to configure a Home Solar Power starter bundle for SunHome. This product converts sun light on the panels into electricity for a small household. The bundle auto-includes two key components:

- 80-100 watt Solar Panels
- 12v @ 105 AH batteries

To find the right quantities of the panels and the batteries required, you must note the electricity usage to calculate the Total daily WattHours and Battery Capacity you require. The bundle also requires the calculated number of wires to be selected where the wiring length can vary based on the site study. The derivation is:

To meet this requirement, you must create:

- *Summary of Usage* attribute group, which contains attributes such as Refrigerator Run Time, Television Run Time, and Microwave Run Time, and receives inputs from the user.
- *Summary of WattHours required* attribute group, which contains attributes that derive the values from the attributes above, using the numeric expression builder.
- SunHome included component option group, which contains options that derive the default quantity from the attributes of the *Summary of WattHours required*, using numeric expression builder.

Using the expression builder, you can create numeric expressions to complete the following tasks.

- Populate product attribute values
- Set the min/max quantity for any option and option group
- Set the default quantity or the min/max quantity for any bundle option

To configure numeric expressions, you must complete the following processes.

Populating product attribute values using numeric expression

Create a product with attributes. You can use these attribute values to create a numeric expressions.

Example

SunHome is a product with the following attributes:

- Refrigerator Run Time
- Microwave Run Time
- Television Run Time
- House Lights

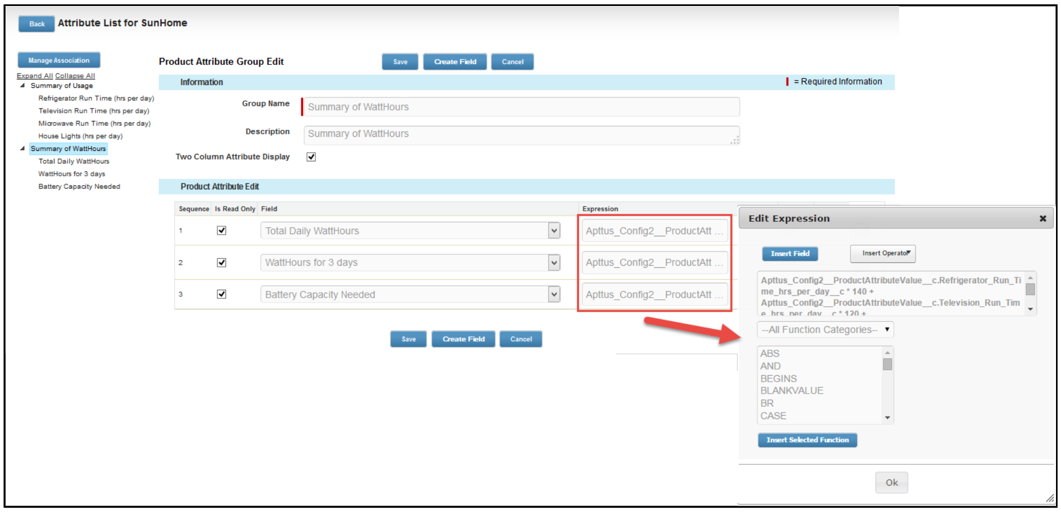
Each attribute has the datatype defined as number. You must also create an attribute group named *Summary of WattHours required* and assign it to the product SunHome. The attribute group contains the following attributes.

- Total daily WattHours required
- WattHours required for 3 days
- Battery capacity

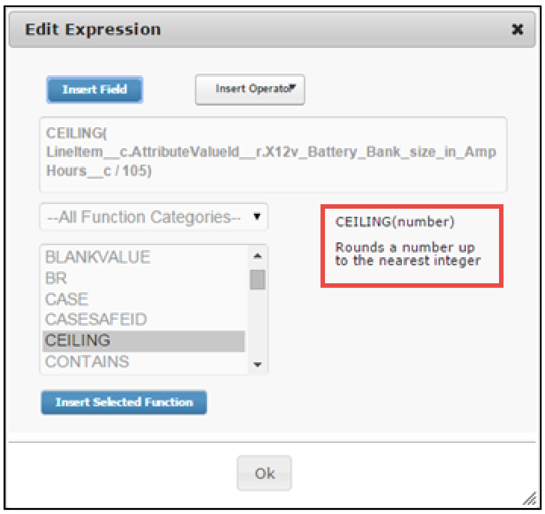
Each attribute has the datatype defined as number. You can derive the values for these attributes using numeric expressions.

To populate product attribute values using numeric expressions

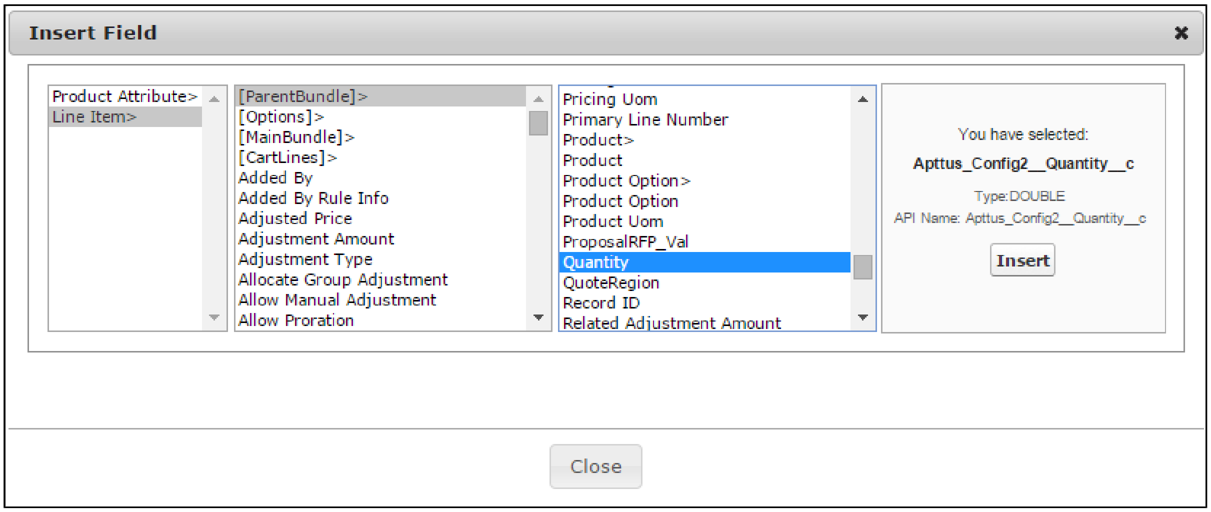
1. From the Products detail page, select SunHome and click **Product Console**.
2. Scroll down to the Additional Data section, and click **Attributes**.
3. Click **Summary of Watthours** required Attribute Group. All the attributes listed under the attribute group are displayed.
4. To associate an expression to an attribute, click on the text field next to the attribute in the Expression Column. An expression builder pop up is displayed.



Selecting any function displays the example syntax on the right on how you can use it.



5. Click **Insert Field**. To retrieve the fields for building the expression, click Insert Field and choose Product attributes or Line Item. Line Item allows you to get access to fields at Product and Quote objects as well.



6. In the Insert Field dialog, click **Product Attribute** and select the required attributes to derive the values for attributes of SunHome. The following expressions are built for each attribute:
   - Total daily WattHours required: Apttus_Config2__ProductAttributeValue__c.Refrigerator_Run_Time_hrs_per_day__c * 140 + Apttus_Config2__ProductAttributeValue__c. Television_Run_Time_hrs_per_day__c * 120 + Apttus_Config2__ProductAttributeValue__c.Microwave_Run_Time_hrs_per_day__c * 1000 + Apttus_Config2__ProductAttributeValue__c.House_Lights_hrs_per_day__c * 200
   - WattHours Required for 3 days: Apttus_Config2__ProductAttributeValue__c.Total_Daily_WattHours__c * 3
   - Battery Capacity Needed (50% Discharge):

     Apttus_Config2__ProductAttributeValue__c.WattHours_for_3_days__c * 0.5

> For more common syntax examples, see Common Examples and Related Syntax For Numeric Expression.

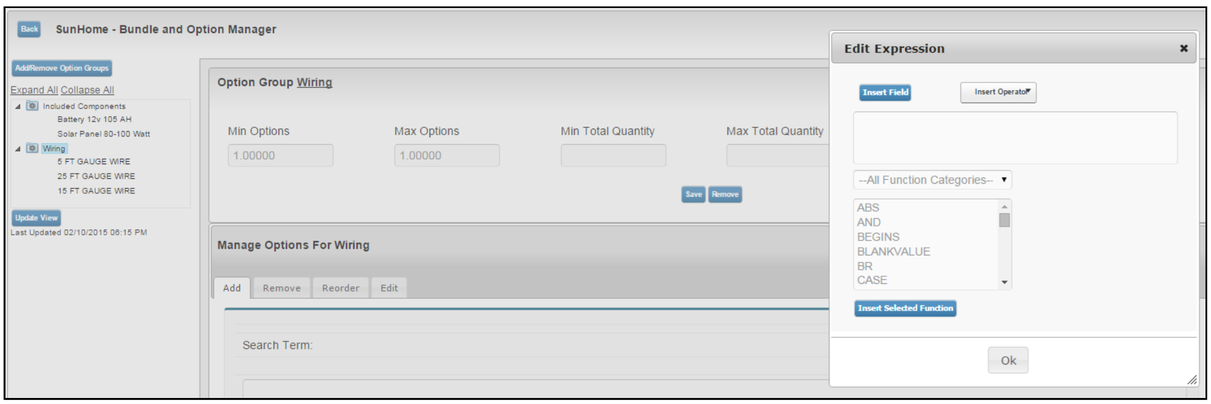
7. Click **Save**.

The expressions are built and the product attribute fields are populated with the value that the expression retrieves.

After completing this process, you must run the Criteria Maintenance job for all expression fields. Refer toRunning criteria maintenance for expression fields.

Configure options for a product.

1. Select the product that you want to assign numeric field expressions to.
2. Click **Product Console**, scroll down to the Catalog section, and click **Manage Bundles /Options**.
3. To associate an expression to min/max total quantity of an option group, select an **Option Group**.



4. Select one or both of the following fields to add an expression.
   - Min Total Quantity
   - Max Total Quantity

   The expression builder opens.
5. Click **Insert Field**. To retrieve the fields for building the expression, click **Insert Field** and choose Product attributes or Line Item. Line Item allows you to get access to fields at Product and Quote objects as well. Ensure that the return value of your numeric expression is of type *Number*.

   For our SunHome Product, we can use this to enforce minimum quantity selection for the option group *Wiring*.

   For more common syntax examples, see Common Examples and Related Syntax For Numeric Expression.

6. Click **Save**.

The expression is built and the option group quantity field is populated with the value that the expression retrieves.

> After completing this process, you must run the Criteria Maintenance job for all expression fields. Refer to Running criteria maintenance for expression fields.

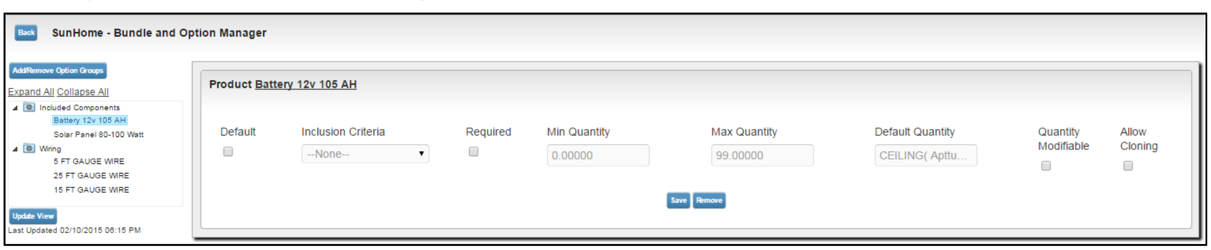
Populating option default and min/max quantity using numeric expression

You must have an existing option product. You must also create an attribute group named *Summary of WattHours required* for SunHome. Following is a list of attributes in the group:

- Total daily WattHours required
- WattHours Required for 3 days
- Battery capacity

Each attribute has the data type defined as number. You must derive the values for options using numeric expressions. You must also create an option group with the options. Based on our scenario, this section outlines the procedure for an option group SunHome Component Quantity with options, Number of Solar Panels and Number of Batteries whose values are derived from the attributes above.

1. Navigate to the SunHome product, and click Product Console.
2. From the Catalog section, click Manage Bundles/ Options.
3. To associate an expression to an option, expand the Option Group, select an Option for which you want to add a field expression.



4. To add an expression, click one or more of the following:
   - Min Quantity

- Max Quantity
- Default Quantity

The expression builder pop up appears.

5. Click Insert Field. To retrieve the fields for building the expression, click Insert Field and choose Product attributes or Line Item. Line Item allows you to get access to fields at Product and Quote objects as well.
6. To derive the values for the options of SunHome, in the Insert Field dialog, navigate to Product Attribute. The expressions built for each of the options are as follows:
   - Battery:

     CEILING(Apttus_Config2__LineItem__c=>ParentBundle.
     Apttus_Config2__AttributeValueId__r.Battery_Capacity_Needed__c/(12 * 105))
   - Solar Panel:

     CEILING(Apttus_Config2__LineItem__c=>ParentBundle.
     Apttus_Config2__AttributeValueId__r.Total_Daily_WattHours__c/(90 * 5))
     Total_daily_WattHours_required__c, 0) / (90 * 5))

     For more common syntax examples, see Common Examples and Related Syntax For Numeric Expression.
7. Click **Save**.

The expressions are built and the option quantity field is populated with the value that the expressions retrieve.

After completing this process, you must run the Criteria Maintenance job for all expression fields. Refer toRunning criteria maintenance for expression fields.

Running criteria maintenance for expression fields

You must have existing field expressions.

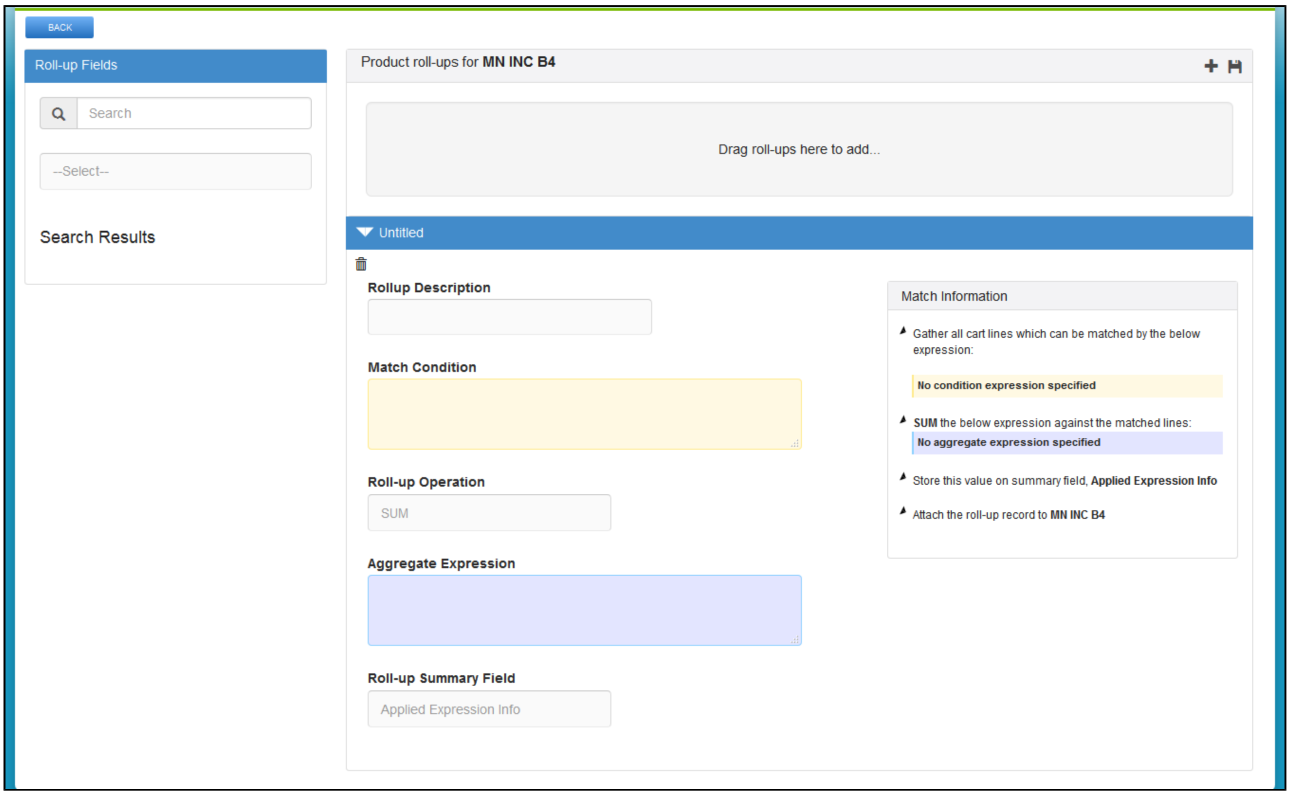
1. Click ✚ and click **Criteria Maintenance**.
2. Click the **Update Expression Fields** button.

This will take some time to update all Expression Criteria fields.

Go to the page where you have set up the expressions to work.

Creating Numeric Roll-Up Summary Fields for Objects

A Roll-up Summary Field can be defined at the Cart level or at a Bundle Product Level to aggregate value from each line item in the given context and with a match condition to filter out unwanted lines.



The **Match Condition** area enables you to enter an Expression created using the Expression Builder that establishes which line items will be filtered and have a Roll-Up.

The **Operator** defines the operation performed on all the aggregated line item values.

The **Aggregate Expression** is an Expression created using the Expression Builder that specifies the value to be rolled-up.

The Roll-up is by default created at the Cart level unless you add the following to the Match Condition:

ParentLineNumber = $scope.LineNumber

This retrieves all the options for the bundle line in the aggregation. You can use this expression to aggregate values at an individual bundle level in order to drive a bundle configuration rule.

The Expressions can be built for any line item field, product field, and attribute value fields. These Roll-Up Fields can be further used for multiple functions such as building other Expressions, Assign default quantities, attributes values.

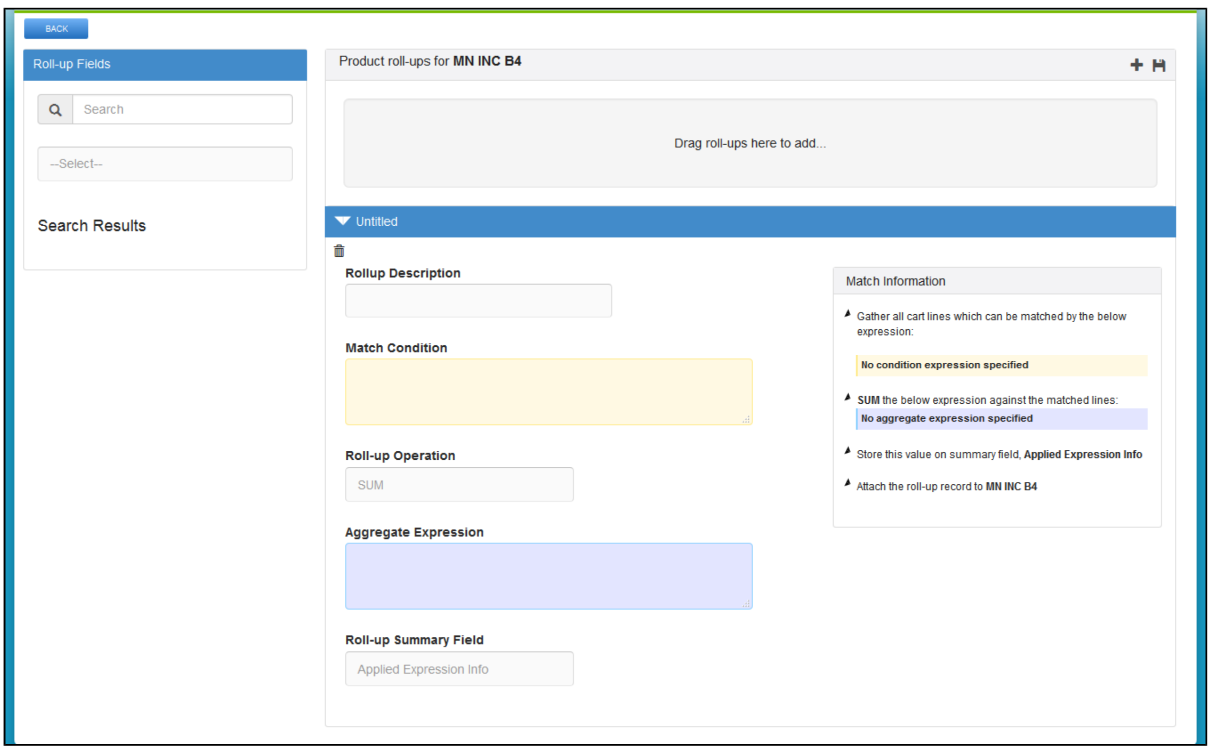
Roll-up Field Name	Return Type	Roll-Up Operation	Match Condition	Aggregate Expression	Object
Total Quantity	Number	SUM	IsPrimaryLine = TRUE	Quantity	Line Item
Total Weight	Number	SUM	Product.Type = Hardware	Quantity * Product. UnitWeight	Line Item
Bundle Heat Dissipation	Number	SUM	ParentLineNumber = $scope.LineNumber && IsPrimaryLine = TRUE	Quantity * Product. UnitHeatDiss	Line Item
Lowest Setup Fee	Currency	MIN	ChargeType = ‘Setup Fee’	ListPrice	Line Item
Max Line Item Discount	Percentage	MAX	ALL	NetAdjustment	Line Item
Avg Cost	Currency	AVG	ExtendedCost>0	ExtendedCost	Line Item

Extensive use of rollups at quote level may have performance implications for a cart with more than 20 product line items. It is also recommended to limit numbers of quote level rollups to below 20.

To create roll-up summary field

Create a custom Salesforce field in Roll-up Summary Object. Ensure that the field is unique and multiple expressions are not used to populate the same field. This field comprises or holds the value derived from the aggregate expression in the Field Expression builder.

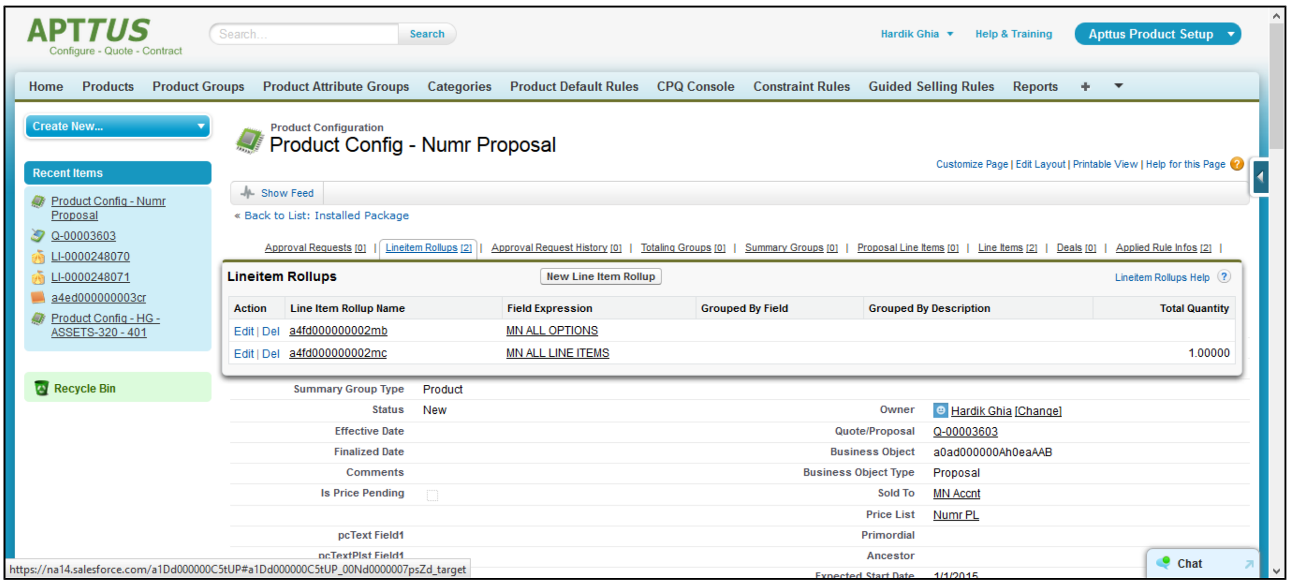
1. You can invoke the field expression builder for:
   - Bundles: To define a rollup field for a particular bundle, navigate to the Products tab, search and select a product and click Product Console.
   - Quote or Product Groups: To define a rollup field for a product group or at header level, navigate to the CPQ Console tab.
2. Click **Rollups** from the Additional Details section.
3. Click ➕ to add a new rollup field for a product.
4. Expand the Untitled section by clicking the right-facing arrow.



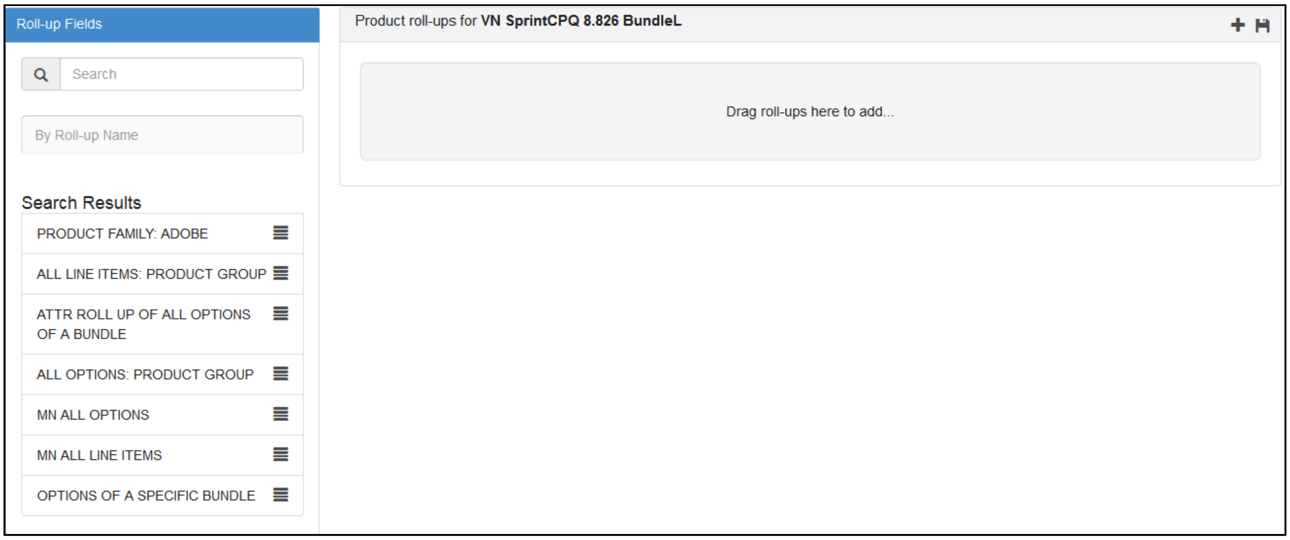
5. Select the Rollup Summary field created on the Rollup Summary object.
6. In the Match Conditions area, specify the expression based on which the results are filtered. The numeric expression rollup is applicable only for those line items that satisfy the criteria in the Match Conditions area.

7. From the Roll-up Operation area, select an operation based on which the rollup is created.
8. In the Aggregate expression area, specify the expression based on which the value is populated in the Rollup Summary Field. For example, (Quantity__c * ProductId_r.Weight__c) *2 Based on the Aggregate expression values, the rollup summary field is populated. Data is populated in The Match Information area based on the information you enter in all the above steps.
9. Click the **Save** icon in the top right corner of your screen.

The rollup for a product is created successfully. You can navigate to the cart page and check whether the rollup is applied successfully.



After you create rollup fields for a product, you can re-use the rollups across products or within the same product by searching for the rollup, product, or product group name in the Search field. Drag and drop the requisite roll-up conditions that you want to use to the right greyed out pane.



Common Examples and Related Syntax For Numeric Expression

This section lists out the common examples and related syntax that you can refer to for your specific scenario.

- Syntax to fetch quantity of any product line selected for the quote:

  ```
  Apttus_Config2__LineItem__c.$CartLines['Apttus_Config2__ProductI
  d__r.Name','Product A'][0].Apttus_Config2__Quantity__c
  ```

  Insert Field > Line item > CartLines > Select the field and provide the filter and index.

- Syntax to fetch sibling option quantity within the same parent bundle:

  ```
  Apttus_Config2__LineItem__c.$ParentBundle.$Options ['Apttus_Conf
  ig2__OptionId__r.Name','Solar Panel 80-100 Watt'][0].
  Apttus_Config2__Quantity__c
  ```

  Insert Field > Line item > ParentBundle > Options > Select the field and provide the filter and index.

- Syntax to retrieve product field value as part of an expression:
  - Retrieving a bundle product field

```
Apttus_Config2__LineItem__c.Apttus_Config2__ProductId__r.
Numr_Numb__c
```

Insert Field > Line item > Product > Select the field

- Retrieving an Option product field when the expression is created for a bundle product.

```
Apttus_Config2__LineItem__c.$Options[field,value][index].
Apttus_Config2__OptionId__r.Numr_Numb__c
```

Insert Field > Line item > ParentBundle > Options > Select the field and provide the filter and index**.**

- In Bundle-In-Bundle setup, if you want to update an option product's attribute, min quantity, and max quantity based on a field value from its bundle product which is an option of another bundle.

```
Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__OptionId__r.Numr_Numb__c
```

In Bundle-In-Bundle setup, if you want to update an option product's attribute, min quantity, and max quantity based on a field value from the top bundle product.

```
Apttus_Config2__LineItem__c.$MainBundle.
Apttus_Config2__ProductId__r.Numr_Numb__c
```

- Syntax to retrieve quote header field as part of an expression:
    - Displaying the Quote header field from a bundle product.

```
Apttus_Config2__LineItem__c.
Apttus_Config2__ConfigurationId__r.
Apttus_QPConfig__Proposald__r.Numr_Field1__c
```

    - Displaying the Quote header field from an Option product.

```
Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__ConfigurationId__r.
Apttus_QPConfig__Proposald__r.Numr_Field1__c
```

- Syntax to retrieve product attribute value as part of an expression:

```
Apttus_Config2__ProductAttributeValue__c.
Total_Daily_WattHours__c
```

Insert Field > Product Attributes > Select the field

- Syntax to define the value of an attribute using another attribute of the same product or option

```
Apttus_Config2__ProductAttributeValue__c.Length__c * .25
```

- Syntax to retrieve parent bundle product attribute value as part of an expression:

```
Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__AttributeValueId__r.Battery_Capacity_Needed__c
```

Insert Field > Line item > ParentBundle > AttributeValue > Select the field

- Syntax to define rollups on header

  Specify the Object to be updated and the field of the object to be updated. In the value expressions field, specify the fields from which you want to fetch the values from.

```
BLANKVALUE($.AttributeValueId__r.
Total_daily_WattHours_required__c, 0) * 3
```

- Syntax to define rollups for Products
  Specify the scope as Product, and enter Product name in the Product field. Ensure that you define a parent field as LineItemId_c in the Rollup Criteria area.
- Syntax to define rollups for Options

  Specify the scope as Product, and in the Match Expression area, enter ParentBundleNumber__c == $condition.PrimaryLineNumber__c

- Syntax to define rollups Product Group

  Specify the scope as Product Group, and enter Product Group name in the Product Group field. Ensure that you define a parent field as LineItemId_c in the Rollup Criteria area.

- Syntax to define rollups for specific Quote Numbers or Account Numbers:

  Specify the scope as Header, and in the Match Expression area, enter

```
BLANKVALUE(ConfigurationId__r.Proposald__r.
Apttus_Proposal__Account__r.Name == 'Account Name',0)
```

- Syntax to define option rollups for specific bundles or multiple bundles

  Specify the scope as Product, specify the product name in the Product field and in the Match Expression area, enter

```
ParentBundleNumber__c == $condition.PrimaryLineNumber__c
```

  Ensure that you define a parent field as LineItemId_c in the Rollup Criteria area.

- Syntax to define a rollup based on the sum of attribute values

  Specify the scope as Product, specify the product name in the Product field and in the Expression Criteria area, specify the Value Expression as

```
BLANKVALUE($.AttributeValueId__r.PerfAttr1__c,0) + BLANKVALUE($.
AttributeValueId__r.PerfAttr2__c,0) + BLANKVALUE($.
AttributeValueId__r.PerfAttr3__c,0)
```

  Ensure that you define a parent field as LineItemId_c in the Rollup Criteria area.

- Syntax to define numeric roll up summary record in a numeric expression

  In order to roll up the specific product attribute (which is numeric) based on the options product quantity roll up, do the following.

  In the Quantity roll up for options,

  Under Match Condition, specify Apttus_Config2__LineItem__c.
  Apttus_Config2__ParentBundleNumber__c != NULL()

Select SUM as roll up operation.

Under Aggregate expression, specify Apttus_Config2__LineItem__c.
Apttus_Config2__Quantity__c

Then use the roll up summary in the expression in the following way:

```
Apttus_Config2__LineItem__c.$Rollups['fieldName',value].
Apttus_Config2__TotalQuantity__c
```

- Syntax to set attribute value of a product from the attribute value of another product in the cart:

```
BLANKVALUE(VALUE(Apttus_Config2__LineItem__c.$CartLines['Apttus_
Config2__ProductId__r.Family','Laptop'].
Apttus_Config2__AttributeValueId__r.Size__c),17)
BLANKVALUE(VALUE(Apttus_Config2__LineItem__c.$CartLines['Apttus_
Config2__ProductId__r.Name','Toshiba Satellite S875-S737617.3
Inch'].Apttus_Config2__AttributeValueId__r.Size__c),14)
BLANKVALUE(VALUE(Apttus_Config2__LineItem__c.$CartLines['Apttus_
Config2__ProductId__r.ProductCode','LP104'].
Apttus_Config2__AttributeValueId__r.Size__c),14)
```

- Syntax to set attribute value of an option item with the attribute value of another option item inside the same bundle.

```
 Apttus_Config2__LineItem__c.$ParentBundle.$Options['Apttus_Conf
ig2__OptionId__r.Name','Option 1.1'].
Apttus_Config2__AttributeValueId__r.Length__c
```

- Syntax to set Option's Default/Min/Max Quantity based on attribute value of the parent bundle:

```
BLANKVALUE(VALUE(Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__AttributeValueId__r.
Data_Transfer_Limit_Month_in_TB__c),0)
BLANKVALUE(Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__AttributeValueId__r.Number_of_Users__c+5, 0)
```

- Syntax to conditionally set the value of attribute or min/max/default quantities of an option item, depending on the value of an attribute from the parent bundle.

```
IF(Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__AttributeValueId__r.Edition__c=='Enterprise',
BLANKVALUE(Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__AttributeValueId__r.Number_of_Users__c+5, 0),
BLANKVALUE(Apttus_Config2__LineItem__c.$ParentBundle.
Apttus_Config2__AttributeValueId__r.Number_of_Users__c+10, 0))
```

- Syntax to set the quantity of an option using quantity of another option in the same bundle.

```
 BLANKVALUE(Apttus_Config2__LineItem__c.$ParentBundle.$Options['
Apttus_Config2__OptionId__r.Name','Option 1.1'].
Apttus_Config2__Quantity__c * 2, 0)
```

- Syntax to set the value of an attribute value or min/max/default quantities using a numeric rollup:

```
 BLANKVALUE(Apttus_Config2__LineItem__c.$Config.$Rollups['Name',
'Cart Weight'].Total_Weight__c,0)
```

- Syntax to set quantity of a product to the sum of quantities of other 2 products in the cart.

```
BLANKVALUE(Apttus_Config2__LineItem__c.$CartLines['Apttus_Config
2__ProductId__r.Name','Logitech Keyboard K120'].
Apttus_Config2__Quantity__c,0) +
BLANKVALUE(Apttus_Config2__LineItem__c.$CartLines['Apttus_Config
2__ProductId__r.Name','Logitech Wireless Keyboard'].
Apttus_Config2__Quantity__c,0)
```

Price Management

Managing Price Lists and Price List Items

Pricing enables you to rapidly set up pricing structures. Pricing comprises of price lists and price list items. A price list controls which products are visible to the end user. A price list contains several price list items; each linked to a product.

You can simply select options and have them priced completely accurately reflecting the latest price lists and discounting rules. You can price any products or bundles based on features or options selected. You can set up pricing rules, constraints, dependencies and extraneous variables.

- Creating a Price List
  Price lists are containers of price list items. Price lists can also have a currency set, if your environment is multi-currency enabled.
- Associate Price List with Categories
  Price list must be associated to an Offering category for the products to be displayed on the catalog.
- Creating Price List Items
  Price List Items contain detailed pricing information about a product.

Creating a Price List

Price lists are containers of price list items. Price lists can also have a currency set, if your environment is multi-currency enabled.

Price lists are associated with categories so that an end user can view and select products from the shopping cart. For example, Price list A is associated with Category B, when you select price list A on the Quote/Proposal Edit page, only category B and associated products are displayed on the shopping cart.

You can create a price list at any time; however, it is most efficient to create a price list before creating products to easily associate the products to the price list.

To create a price list

1. Navigate to the CPQ Console tab and from the Price Management section, click Add Price Lists.
2. Type a mandatory Price List Name and type a Description.
3. To make the price list active, select Active.
4. Click Save.

A price list is created and saved.

You can associate all related pricing, products, and options with them.

Associate Price List with Categories

Price list must be associated to an Offering category for the products to be displayed on the catalog.

The related list against the price list displays the category associated to it and this category is displayed to the end user in the shopping cart. In this way, you can restrict the display of products by category.

To associate price list to a category

You must have an existing offering category and a price list.

1. Navigate to the CPQ Console tab and from the Product Management section, click Manage Categories.
2. Select a category to associate with a price list, scroll down to the Price Lists related list, and click New Price List Category to find and associate the appropriate price list.
3. Enter a mandatory Price List, and enter a Hierarchy.
4. Click **Save**.

The price list is associated with a category.

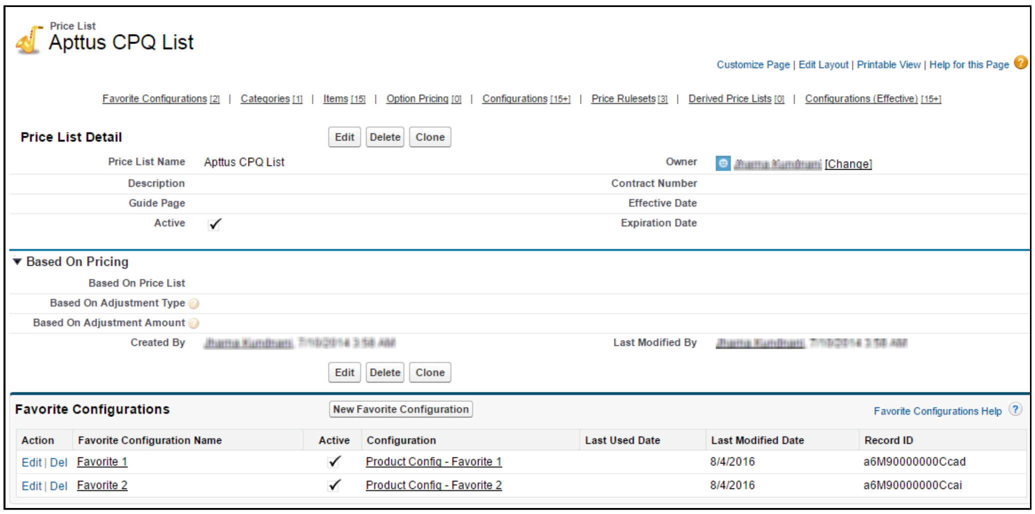
Viewing and publishing Favorite Configurations for a Price List

After the Sales rep saves the favorite configurations, as an admin, you can view these favorite configurations collectively under a Price List. The favorite configurations are always linked to the Price List and after you associate this Price List to your Quote/Proposal, a separate category, named **Favorites**, containing your favorite configurations is displayed on the Catalog page.

You can do the following customizations on favorite configurations:

- Associate an image
- Customize the label of your favorite category
- Customize the fields that should be displayed to the sales reps on the catalog page
- Customize the columns to be displayed on the **Save as My Fav** dialog box on the cart.
- Disable the favorite configuration related UI elements.

For more details, see Catalog Page settings.



The **Publish** button on your favorite configuration record allows y ou to publish a favorite configuration record to another price list. This is useful when you want to use the same configuration records across a number of price lists.

Ensure that you have configured Lookup Field Settings with the values as described in the image below before publishing your favorites to another price list.

Creating Price List Items

Price List Items contain detailed pricing information about a product.

A product can be set up with one or more price list items. Each price list item includes the list price for the product including details of charges such as per unit, flat price, one time, recurring, and more. All the fields on a price list item are price related and impact the final price on the shopping cart. Each price list item represents different ways, a customer is charged for a product. For example: License fee, Implementation fee, and more. Price list items are categorized into price lists.

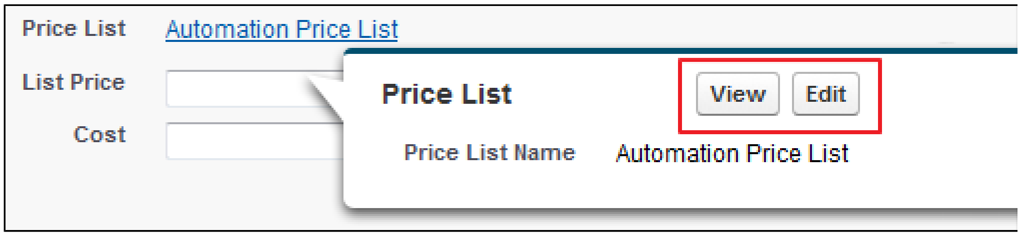
To create a price list item

Navigate to the **CPQ Console** tab.

1. Click **Manage Price Lists** and select an existing price list.
2. Scroll down to the Items related list, and click **New Item**.
3. Enter details in one or more of the following fields:

Option	Description
Product	Enter a product.
Charge Type	Select a charge type. These are a separate, identifiable element of charges tha be used for pricing. For example: Standard Price, Maintenance Fee, Installatio Fee, and more. This is a way to capture multiple types of fees for a product or line item. For instance, a product has a product fee but can also have a separate fee for installation. In such a case, you will create two price list item entries for the sar product and specify the charge type for each one of the two entries. The charg type will be automatically added to the cart when you add the product so that a charges associated with the product are included in the pricing. Apttus does not recommend setting up the bundle and its option to have differe frequencies with the same charge type.
Charge Type Criteria	This is used for conditional charge types that are included only if a certain crite met and as part of adding the main product to the cart. Setting this criteria eith Line Item or Attribute field will only apply this price list item under those conditi For example, this may be a charge type based on workstations; whereas the product may have a charge type based on users.

Option	Description
Price Type	Select *One Time*, *Recurring*, or *Usage*. Selecting *Recurring* or *Usage* enables **Frequency** field. For instance, if you add a subscription product, you will choo type as recurring and then specify the frequency as monthly, yearly, and so on Example: You charge 1$ per month for a term of 1year, the price is calculated accordingly. If you select **Price Type** as *Usage*, you must select **Price Method** as *Tiered R*
Frequency	This is based on your selection from the Price Type field. Select an option to indicate how often the product will be charged.
Price Method	Select one of the following: • Per Unit: Multiplies the price with the quantity. • Flat Price: Applies a flat price. • Percentage: Enables the Percent, Percent Applies To, Related Price Lis fields. • Related Price: Used when the price is calculated as a function of the pri other lines in the same shopping cart. • Tiered Rate: Default price method for Usage Price Type.
Percent, Percent Applies To, Related Price List Item	These allow you to set price dependencies such that one price is a percentage the unit or extended price of another product. Refer Related Pricing for more information.
Description	Type a brief description to specify the purpose of the price list item.
Active	To make the price list item active, select this check box.
Price List	

Option	Description
	Displays the price list. Moving the cursor on the link enables you to view or edi price list. 
List Price	Type a list price for a product in the price list item. If the price list item has matı or may be overridden, enter **0** as the list price.
Cost	Type a cost of a product.
Price Uom (Unit of measure)	Select an option to specify the unit price you are charging is for an hour, day, y and so on. This helps the end user while pricing products in the cart. You can modify this in the price list item object.
Min Price	Enter the minimum price.
Max Price	Enter the maximum price.
Min/Max Price Applies To	Select one of the following: • *Base Price* - This is the derived price for a product when you add it to th cart. This can be same as the list price or different from the list price, if y have defined a price matrix or a price rule that dynamically arrives at the price of the product based on other factors. • *Base Extended Price* - This is derived by Base Price * Quantity * Term.
Effective Date	Enter an effective date for the price list item.
	Enter an expiry date for the price list item.

Option	Description
Expiration Date	

Apttus Pricing Engine does not support custom values for **Price Type**, **Frequency** and **Price Method** at the Price List Item level or in the Pricing Callback classes. Using custom values may cause unexpected system behavior.

4. Click **Save**. A price list item is created for a product or option. The price list items created displays the products and options to the end user in the pricing cart page.

Ensure that the **Charge Type** and **Frequency** for the bundle and its associated option product are same.
Apttus CPQ uses charge types as the consolidation factor for grouping. Such grouping can be at the bundle level, Summary Group level, or Sub-total level.
In this way, we ensure that the line items with the same **Charge Type** and **Frequency** calculate the following fields on the Cart appropriately:

• Selling Term
• Adjustments applied on a bundle line or a Summary group line

- Manually Adjusting Options Pricing for a Bundle
  The **Options** tab display the option product and associated price for a bundle. You can control how the options price rollup to the bundle price. You can also control the display of options adjustments on the pricing cart page.
- Applying Price Matrices from the Matrices Tab
  If the product has a matrix based price, click the Matrices tab.
- Applying Default Pricing to a product
  You can apply default pricing to a product from the Defaults tab.
- Defining Tax and Billing for a product
  You can define the way your product should be billed and taxed from the Tax & Billing tab.
- Applying Miscellaneous Pricing to a Product
  You can apply miscellaneous pricing to a product from the Miscellaneous tab.

- Support for Weekly as Pricing Frequency for a Charge
  Products can now be configured to have recurring charges with a pricing frequency set as weekly. The customer would then be charged based on the number of weeks between the start and end dates of the product.

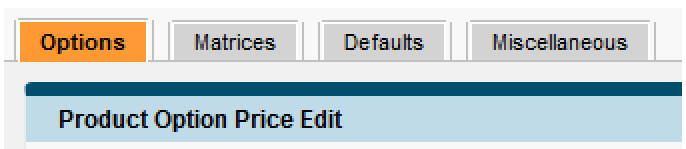
Manually Adjusting Options Pricing for a Bundle

The **Options** tab display the option product and associated price for a bundle. You can control how the options price rollup to the bundle price. You can also control the display of options adjustments on the pricing cart page.

To manually adjust options pricing for a bundle

You must have an existing price list item.

1. Select an existing price list item and click Edit and click the Options tab.



2. Enter information in one or more of the following editable columns:

Column	Description
Adjustment Amount	Enter an amount to adjust the options price.
Adjustment Type	Select an adjustment type.
Price Included In Bundle	Select to include the price with the bundle product. Selecting Price Included in Bundle sets the Adjustment Amount to zero and Adjustment Type to Price Override. This means the option price is included in the bundle price and therefore is free.
Allow Manual Adjustment	Select this to enable the end user to make manual adjustments. Manual adjustments are done through the adjustment type and amount fields on the pricing cart page.

Allocate Group Adjustment	Select this to include the product in a manual group discount, if applied, on the pricing cart page.

3. Click **Save**.

The option price for a bundle is adjusted and saved.

Applying Price Matrices from the Matrices Tab

If the product has a matrix-based price, click the Matrices tab.

A price list item can have more than one matrix evaluated in a sequence. For more information, see Defining or Creating Price Matrices.

In order to work with matrix-based pricing, you must create price dimensions for a product. You can create up to six dimensions, bringing in attributes from the line item or header objects of any data type within Salesforce. The dimension value type has three values: Discrete, Range, and Cumulative Range.

Applying Default Pricing to a product

You can apply default pricing to a product from the Defaults tab. This tab enables you to apply default quantity and pricing to a product.

To apply default pricing to a product

You must have an existing price list item.

1. Select an existing price list item and click **Edit** and click the **Defaults** tab.
2. Enter details in one or more of the following:

Option	Description
Default Quantity	Enter the default quantity.

Option	Description
Default Quantity From	Select an option from where default quantity is to be considered. If the price list item derives its default quantity from a specific attribute, when the user navigates to the catalog page and selects a product and its attribute, the default quantity of the product is derived based on a number selected from an attribute of type picklist or number.
Default Quantity Field	This field is dependent on the Default Quantity From field. Select the Product Attribute using which the default quantity of the product is derived.
Auto Update Quantity	Select to automatically update the quantity on the Cart. This is useful when the quantity of your product is derived using a Numeric Expression and you want to reflect the changes made by the Numeric Expression on the Cart automatically.
Auto Cascade Quantity	Select to automatically cascade quantity.
Is Quantity Read Only	Select this to make the Quantity field non-editable on the cart.
Default Price From	Select an option from where default price can be considered.
Default Price Field	This field is dependent on the Default Price From field. After selecting an option from the Default Price From field, select a specific field from where default price can be considered.
Default Selling Term	Enter the default selling term.

Auto Cascade Selling Term	Select to automatically cascade selling term.
Is Selling Term Read Only	Select this to make the Selling Term field non-editable on the cart.

3. Click **Save**. Default pricing is applied and saved.

Applying Miscellaneous Pricing to a Product

You can apply miscellaneous pricing to a product from the Miscellaneous tab.

This tab enables you to do the following:

- You can include an option price in a bundle.
- You can enable price ramps if a product is rampable, and enable commitments.
- You can control if a usage tier can be modified or not.
- You can manually adjust options pricing for a bundle. You can also control how the options price rollup to the bundle price and display of options adjustments on the pricing cart page.

To apply miscellaneous pricing to a product

You must have an existing price list item.

1. Select an existing price list item and click **Edit** and click the **Miscellaneous** tab.
2. Select the check boxes as needed:

Price Included In Bundle	

	Select to include the price with the bundle product. Selecting Price Included in Bundle sets the Adjustment Amount to zero and Adjustment Type to Price Override. This means the option price is included in the bundle price and therefore is free.
Enable Price Ramp	Select this check box to make a product price rampable.
Enable Commitment	Select this to enable commitment.
Is Usage Tier Modifiable	Select this check box to enable an end user to modify the usage tier.
Disable Asset Integration	Select to prevent the creation of asset corresponding to this product during the quote process. For now, you can disable asset creation only for standalone products. It is recommended not to block asset creation because it hinders the preferable system flow of asset creation on quote acceptance. Once you install Apttus Billing Management version 4.97 installed or later, order activation continues as billing now ignores any line without an asset id. You cannot see bills created for those orders because there is no asset.
Allow Manual Adjustment	Select this to enable the end user to make manual adjustments. Manual adjustments are done through the adjustment type and amount fields on the pricing cart page.
Allocate Group Adjustment	Select this to include the product in a manual group discount, if applied, on the pricing cart page.
Allow Proration	Select to allow proration. On the Cart page, to make Selling Frequency editable by the user, this check box must be selected.

3. Click **Save**.

Miscellaneous pricing items applied to a product and saved.

Support for Weekly as Pricing Frequency for a Charge

Products can now be configured to have recurring charges with a pricing frequency set as weekly. The customer would then be charged based on the number of weeks between the start and end dates of the product.

To set charge type frequency as weekly

A valid Price List should exist. If you are upgrading from a previous version follow the Post Upgrade Steps: Add the value Weekly to the picklists of the following objects:

Object	Picklist to add the value to
Price List Item	Frequency (Update the Price Type dependency picklist accordingly)
Asset Line Item	Pricing Frequency, Selling Frequency
Line Item	Pricing Frequency, Selling Frequency
Order Line Item	Pricing Frequency, Selling Frequency
Summary Group	Frequency
Frequency Conversion Rate	From Frequency, To Frequency

1. Create a Price List Item for a Product.
2. Set the **Price Type** as *Recurring*.
3. Set the **Frequency** as *Weekly*.
4. Specify the **List Price** and other requisite fields and click **Save.**

When a product having a charge with frequency as weekly is added to the cart, the term for that charge is calculated based on the number of weeks between the start date and end date. The final price will reflect the term.

Defining Tax and Billing for a product

You can configure all Billing and Tax related information for your product from the Tax & Billing tab.

To define Taxes and Billing

You must have an existing price list item.

1. Select an existing price list item and click **Edit** and click the **Tax & Billing** tab.
2. Enter details in one or more of the following:

Taxable?	Select this check box if the order is a taxable product or service.
Tax Inclusive?	Select the check box if the price of the product is inclusive of taxes.
Tax Code	Lookup to select the relevant tax code.
Billing Rule	Select one of the following options • **Bill in Advance** - To bill your customer before the product is delivered • **Bill in Arrears** - To bill your customer after the product is delivered • **Ready for Billing Date** - To bill your customer on the **Ready for Billing Date** you defined for the Order Line Item • **Payment on Receipt** - To bill your customer when the invoice is presented
Billing Frequency	Select one of the following options • **Monthly** - To generate a bill once every month • **Quarterly** - To generate a bill once every three months • **Yearly** - To generate a bill once every year • **Usage** - To generate a bill based on usage

Minimum Usage Quantity	If you defined the **Billing Frequency** to be usage based, enter a value to set the minimum quantity of billable consumption.
Maximum Usage Quantity	If you defined the **Billing Frequency** to be usage based, enter a value to set the maximum quantity of billable consumption.

3. Click **Save**.

Managing Price Rules

In addition to managing prices for products using price lists and price list items, you can manage prices using pricing rules. Pricing rules manage special offers, discounts, and charges at either the line item level or the summary line item level. Many pricing rules can be associated to a Ruleset and through to products.

Price Rulesets are a mechanism to allow particular families, categories or groupings of products to have either line item pricing adjustments applied or summary pricing adjustments applied. Typical examples of these are volume discounting rules or promotional pricing rules.

- Creating Price Ruleset
  Price Rulesets allow for pricing adjustments across a range of products via price rules. Both line item pricing adjustments and summary pricing adjustments can be applied. For example: Discounting rules, promotional pricing, adjustments in aggregate based on a Product Family, and more.
- Creating Price Rule
  You can associate multiple rules to a ruleset or through to a product. Price matrices can be defined in price rules as well as conditional price rules (rules that use boolean logic such as AND/OR). Price rules can be dimensional, which use similar concept as Price Matrices or Conditional, that relate to fields for pricing adjustments.

Creating Price Ruleset

Price Ruleset allows pricing adjustments across range of products via price rules. Price Ruleset manages special offers, promotions and enables discounting based on the specific product attribute. A price ruleset can contain one or more rule entries and it can be dimensional, which employs a similar concept as Price Matrices. You can apply line item pricing adjustments as well as summary pricing adjustments. For example: Discounting rules, promotional pricing, adjustments in aggregate based on a Product Family, and more. Also, if your customer has Gold Rating, and you want to automatically discount specific line items on a proposal by 30%, you should use Price Ruleset.

To create a price ruleset

You must have an existing price list.

1. Navigate to the **CPQ Console** tab and click **Add Price Rules**.
2. Enter a mandatory **Ruleset Name** and a mandatory **Sequence** in which the system will evaluate multiple rulesets. Typically, you will perform line item adjustments first and then any summary adjustments.
3. Enter details in one or more of the following fields, as required:

Option	Description
Category	Select All, Agreement, or Proposal. This indicates if the ruleset is relevant to Agreements, Proposals, or both.
Application Level	Select a level to which this ruleset will be applied. *Line Item*: Selecting this will apply the line item adjustment within the line item net price in the shopping cart. *Bundle*: Selecting this will apply the adjustment to a bundle and its options and is displayed in the Totals tab in the shopping cart. *Aggregate:* Selecting this enables the **Application Method** field.
Application Method	This field is dependent on the Application Level field. This indicates that you want the adjustment to select products in aggregate but apply an adjustment as a summary line in the Totals tab or spread the adjustment over numerous products. For example: Buy 2 get 1 free.
Ruleset Criteria	Allows you to set criteria for a line item rule or a bundle, depending on the Application level you have selected, such that the ruleset only applies when it satisfies a line level field value or a product attribute value. Click **New** to fill in your criteria.

Option	Description
Stop Processing more Rules	Selecting this indicates that if the ruleset is satisfied, the system will not process any more rulesets.
Active	Select this to set the ruleset as active.
Price List	Select a price list. The ruleset will only source product prices with the selected price list and then apply adjustment criteria.
Product Family	Select the Product Family. This is the Product Family field on the products object. The ruleset will only source product prices with the selected Product Family.
Product Category	Select a category. The ruleset will only source product prices within the selected category.
Product Group	Click to search and select a custom product group to the ruleset will apply the pricing adjustments. These custom product groups have no relation to a category, a Product Family, or any other product designation.
Charge Type	Select a charge type to which the ruleset will apply adjustments.
Effective Date	Not required as long as the price ruleset is Active, but can be used for promotional rules, and more.
Expiration Date	Select an expiration date.
Description	Enter a description for the ruleset.

4. To save the ruleset, click **Save.**

By filling out this criteria, the source products and prices are then designated and the ruleset can apply adjustments via price rules.

Creating Price Rule

You can associate multiple rules to a ruleset or through to a product. Price matrices can be defined in price rules as well as conditional price rules (rules that use boolean logic such as AND/OR). Price rules can be dimensional, which use similar concept as Price Matrices or Conditional, that relate to fields for pricing adjustments.

One ruleset can have multiple rules and are evaluated in order of the rule sequence. A rule can be dimensional or conditional. A price rule determines the actual price adjustment made.

To create a price rule You must have an existing Price Ruleset.

1. Navigate to the **CPQ Console** tab and click **Manage Price Rules**.
2. Select an existing price ruleset, scroll down to the **Rules** related list, and click **New Rule**.
3. In **Rule Name**, enter a mandatory rule name.
4. In **Sequence**, enter a mandatory sequence.
5. From **Rule Type**, select *Dimension* or *Condition*. In setting a dimensional price rule the process is exactly the same as creating a price matrix for a product.
6. From **Adjustment Applies To**, select an option to which the adjustment is applied. This is for line item adjustments only.
7. From **Adjustment Charge Type**, select an option to set summary level adjustments as a charge type. You can change this in the Price Rule object.
8. Selecting **Stop Processing More Rules** indicates that if the ruleset is satisfied, the system will not process any more rulesets.
9. From **Allowable Action**, select an appropriate action for adjustments.
10. To allow complete removal of a price adjustment, select **Allow Removal of Adjustment**.
11. In **Description**, enter a description for the rule.
12. To make the rule active, select **Active**.
13. Do one of the following:
    - To save the rule, click **Save**.
    - To save and create a new rule, click **Save & New**.

Defining or Creating Price Matrices

Price Matrices are an advanced pricing concept used to define tiered pricing paradigms, or complex pricing structures with multiple criteria. Common examples are pricing tiers for a product based on user count or particular customer or transactional dimensions.

- Creating Price Dimensions
  Price Dimensions are an important concept in order to link any field within Salesforce into the pricing tables. Any field that will determine a pricing adjustment such as Quantity, Term, and more, needs a dimension created for it.
- Creating Price Matrices
  These are an advanced pricing concept used to define tiered pricing paradigms, or complex pricing structures with multiple criteria. Common examples are pricing tiers for a product based on user count or particular customer or transactional dimensions. A price list can have more than one price matrix.

Creating Price Dimensions

Price Dimensions are an important concept in order to link any field within Salesforce into the pricing tables. Any field that will determine a pricing adjustment such as Quantity, Term, and more, needs a dimension created for it.

To create price dimensions

1. Navigate to the Price Dimensions tab and click New.
2. In Dimension Name, enter a mandatory name for the dimension.
3. From Context Type, select one of the following:

Option	Description
Line Item	Allows you to choose any field on the line item shopping cart. For example: Quantity, Term, and more.
Product Attribute	Allows you to choose any product attribute that has been associated to a product via an attribute group. From Attribute, click to select an attribute group.

Option	Description
Formula Field	Allows you to create a formula field containing the syntax as per the custom syntax below. You can define a syntax once and can reuse it across multiple price dimensions. There may be scenarios where pricing varies based on Customer Priority status. For example, a customer can have three statuses Gold, Silver, and Platinum. You can define a pricing dimension based on the status of the customer. The Customer Priority field is a custom field you define in the opportunity. You can create a formula field on the *Apttus_Config2__LineItem__c* object referencing the Customer Priority field. You can later define a pricing dimension for the same formula field and use it within the rulesets to define different pricing tiers for each customer priority. Even though the same Customer Priority field is available in the Line Item context type as well, in scenarios where you have a large number of line items defined with different data types, searching and setting specific values for the context type can be cumbersome. Searching and setting a formula field context type within Apttus_Config2__LineItem__c filters only the fields having formula as a datatype.
Custom	Allows you to define a custom syntax for the system to retrieve a fields value. It is most commonly used to retrieve Quote or Agreement Header fields. This is always from the context of the Line Item, so the format for the fields must be: For Quote Header fields: Apttus_Config2__ConfigurationId__r.Apttus_QPConfig__Proposald__r. INSERTYO URFIELDAPINAMEHERE -or- For Agreement Header field: Apttus_Config2__ConfigurationId r.Apttus_CMConfig__AgreementId__r. INSERTYOURFIELDAPINAMEHERENote: The field must be the fields API name and not the field label or name.

4. In Description, enter a decription to specify the purpose.

5. Do one of the following:
    - To save the price dimension, click Save.
    - To save and create a new price dimension. Click Save & New.

Creating Price Matrices

These are an advanced pricing concept used to define tiered pricing paradigms, or complex pricing structures with multiple criteria. Common examples are pricing tiers for a product based on user count or particular customer or transactional dimensions. A price list can have more than one price matrix.

To create price matrices

You must have existing price dimensions.

1. Navigate to the Matrix Pricing related list from within an individual price list item.
   -or-
   Navigate to the Matrices tab under the fields for the new price list item details.
2. In **Sequence**, enter the sequence number.
3. From **Matrix Type**, select *Dimension*.
   If you choose *Condition*, note that the option or attribute pricing at the 1st level sub-bundle is not supported.
4. From **Dimension**, select a price dimension. You can enter up to six dimensions, which bring in attributes from the line items or headers of any data type within Salesforce.
5. From **Dimension Value Type**, select one of the following:

Option	Description
Discrete	The system only considers the specific matrix values.
Range	The system considers the matrix values as inclusive of the numbers in the range.
Cumulative Range	The values will be evaluated cumulatively. For example: If quantity = 15 and price is $1 for a range of is 0-10, and price is $2 for a range of 11-15, then 10x$1+5x$2 = $20 will be the price of the product.

6. To stop the system from evaluating further matrices if a match is found within this matrix, select the **Stop Processing More Matrices** check box.
7. In **Description**, enter a description.
8. Click the **Details** tab, to enter the details for the price matrix.
9. In the **Quantity** column, enter a value. The values are inclusive. For example, the first line here indicates a range of 0-10. The second line indicates a range of 11-20, etc. All adjustments are adjusting the price list item-list price. To account for an infinite number of values, set the quantity to 999999, which will pull a range of “infinity.” In this example, setting 999999 indicates the range 41-999999.
10. In the **Adjustment Amount** column, enter a percentage or a number.
11. From **Adjustment Type**, select the kinds of adjustment required, if the criteria is met. An example is an override to a list price, markup, or discount. You can add more than one matrix rows by clicking ✚ and delete rows by clicking ▬ . You can also move a row up or down by clicking ▲ or ▼ .
12. Do one of the following:
    - To save a price matrix, click **Save**.
    - To save and create a new price matrix, click **Save & New**.

Pricing Callback Classes

The pricing callback is invoked during pricing action and executes the various methods in three modes: BASEPRICE, ADJUSTMENT, ROLLDOWN.

Pricing Callback class enables you to add pricing logic to your cart that cannot be achieved by out of the box pricing mechanisms, such as Price Rulesets and Price Matrices. The Pricing Callback is executed for each bundled product or standalone in the cart. A separate transaction call is initiated for each product in the cart. The Pricing Callback is invoked when you are on the cart page.

For a pricing callback class ensure that you define the following methods:

- **start() method**: Is the first method executed when a pricing callback is invoked.
- **setMode() method**: Specifies the mode of the callback. Based on the mode of the callback you can define different business logic. The modes are:
    - ADJUSTMENT: The mode of the call back when adjustments are made to the cart and you click **Reprice**.

    - ROLLOUT: The mode of the callback when Apttus_Config2.CustomClass. PricingMode.BASEPRICE == mode)) .
    - BASEPRICE: The mode is base price when you click **Go to Price**.
- **beforePricing() method**: Using this method, you can define the logic for net price calculation. In beforePricing, no pricelist item is associated to the cart line items. Price Matrix and Price Ruleset calculation occurs when the mode of the callback is Base Price Mode.
- **afterPricing() method**: In this method, define the logic to be executed after Net Price calculation, that is after the system has already calculated Net Price when this method is invoked.
- **finish() method**: In this method, the logic for final processing is defined. This is the last stage of the pricing callback.

A new method *getLineItemCollsInCurrentBatch()* is introduced with the latest release which is used to fetch a collection of line item records in the current pricing batch. Each line item collection indicates a bundle or standalone product. The collection of line item records is stored in a map and can be used in the before or after pricing methods inside a Pricing Callback class.

You can use this method with the cart object inside the start method.

Note: This method is available only in the BASEPRICE mode.

BASEPRICE mode is used primarily to set quantity, term and so on in the line item beforePricing() method. Base price mode is called for each line number. If there are two line numbers, the BASEPRICE mode would be invoked twice. The following methods in the callback are invoked during the BASEPRICE mode:

- start
- setMode
- beforePricing
- onPriceItemSet
- afterPricing
- finish

The line item collection in beforePricing may not match the line item collection in afterPricing if the product has multiple charge types. In this case, the afterPricing method would have more line items. Any modification to custom fields need to be done in the beforePricing method.

ADJUSTMENT mode is invoked for all items (including options) in the cart after applying adjustments to compute **Net Price**. Only the adjusted price or the adjustment amount may be changed at this point. The following methods in the callback are invoked during the Adjustment mode:

- start
- setMode
- beforePricing
- beforePricingLineItem
- afterPricingLineItem
- afterPricing
- finish

ROLLDOWN mode is invoked to apply the total or group level discounts at the line level. The group discounts are distributed across the bundle or standalone lines. The following methods are invoked during the ROLLDOWN method.

- start
- setMode
- beforePricingLineItem
- afterPricingLineItem
- finish

Depending on the discount applied at the group level, the beforePricingLineItem and afterPricingLineItem may be applied at the option lines after the execution of the methods at the bundle lines level.

When you define a pricing callback, ensure that you verify the following:

- Implement the latest interface version 3 for your pricing callback class.
- Choose the correct mode for your custom logic because a ll or some of the methods in the class will be called during each mode.
- Most widely used modes are either BASEPRICE or ADJUSTMENT. Apttus recommends to avoid using ROLLDOWN mode.

- It is recommended to have logic in the BASEPRICE mode only, in the beforePricing, beforePricingLineItem, afterPricing and afterPricingLineItem methods. ADJUSTMENT mode is exception and should be reviewed with the Product team (PM, PST, Engg) before approval for implementation.
- ADJUSTMENT mode does not have all option items in Large Cart mode and hence it is preferable not to use this mode.
- Any DML statement is not defined. For example, if you want to compute the **Base Price** which is based on some custom logic, assign the final calculated value to **Quantity**, **Selling Term** or any custom field on line item.
- If your logic is dependent on **BasePrice** calculation, then make sure to use **BasePrice Override** field of Line Item. You can put the newly computed **BasePrice** in this field and then in next mode, the system will take the value of **BasePrice** from override field to compute **ExtendedPrice** and **NetPrice**.
- When you do a **Reprice** on shopping cart and change any field on line item, which is used in Price Matrix or Price Ruleset, then BASEPRICE mode will get called along with ADJUSTMENT and ROLLDOWN modes. If you change fields on line item which are not impacting Base Price Calculations (e.g. Giving 10% discount on a bundle line item), then only ADJUSTMENT and ROLLDOWN modes are called.
- Option Line Item are not be called in beforePricingLineItem() or afterPricingLineItem() methods in any mode. If you need to work on an option line item, then do it in beforePricing() or afterPricing() methods.
- If you want to write any custom logic that depends on price list item fields, then you should use onPriceItemSet() method. In this method, the system determines the corresponding price list item for a line item.
- If you have queries in callback class, it is advisable to do it in start() method so that it is not repeated for each line item in the before/after methods.
- Use of updatePrice() method should be avoided unless there is approval from the Product team.

```
/**
 * Apttus Config & Pricing
 * DefaultPricingCallback2
 *
 * @2011-2014 Apttus Inc. All rights reserved.
 */
```



```
global with sharing class SamplePricingCallback2 implements
Apttus_Config2.CustomClass.IPricingCallback3 {

   private Apttus_Config2.ProductConfiguration contextCart;
   private Apttus_Config2.CustomClass.PricingMode currentMode;

  /**
  * Callback at the beginning of the pricing call.
  * Use the start method to initialize state
  * @param cart the cart object
  */
  global void start(Apttus_Config2.ProductConfiguration cart) {
  this.contextCart = cart;
  }

  /**
  * Callback to indicate the pricing mode
  * @param mode the pricing mode
  */
  global void setMode(Apttus_Config2.CustomClass.PricingMode mode) {
  this.currentMode = mode;
  }
  /**
  * Callback after the price list item is set on the given line item
  * @param itemSO the price list item associated with the line item
  * @param lineItemMO the line item
  */
  global void onPriceItemSet(Apttus_Config2__PriceListItem__c itemSO,
Apttus_Config2.LineItem lineItemMO){
  }

  /**
  * Callback before pricing the line item collection
  * Use this method to do all required pre-processing to prepare the
line items for pricing.
  * @param itemColl the line item collection to pre-process
  */
  global void beforePricing(Apttus_Config2.ProductConfiguration.
LineItemColl itemColl) {
  }

  /**
  * Callback before pricing the given line item in the line item
collection
  * Use this method to do all required pre-processing to prepare the
line item for pricing.
  * @param itemColl the line item collectionholding the line item
```



```
 * @param lineItemMO the line item to pre-process
 */
 global void beforePricingLineItem(Apttus_Config2.
ProductConfiguration.LineItemColl itemColl, Apttus_Config2.LineItem
lineItemMO) {
 }

 /**
 * Callback after pricing the given line item in the line item
collection
 * Use this method to do all required post-processing after the line
item is priced
 * @param itemColl the line item collection holding the line item
 * @param lineItemMO the line item to post-process
 */
 global void afterPricingLineItem(Apttus_Config2.
ProductConfiguration.LineItemColl itemColl, Apttus_Config2.LineItem
lineItemMO) {
 }

 /**
 * Callback after pricing the line item collection
 * Use this method to do all required post-processing after line
items are priced.
 * @param itemColl the line item collection to post-process
 */
 global void afterPricing(Apttus_Config2.ProductConfiguration.
LineItemColl itemColl) {
 }

 /**
 * Callback after all batches of line items are processed
 * Use the finish method to release state
 */
 global void finish() {
 }

}
```

To add call back classes in Custom Settings

1. Go to **Setup > Develop > Custom Settings** and click **Manage** next to **Config Custom Classes**.

2. Click **New**.
3. Fill in the information next to the call back class fields.

If your implementation is using custom callback class, refer to the upgrade information.

Related Pricing

Related pricing is used when you want to derive the price of a Line Item depending on the price of another Line Item. The Line Item can be of a Standalone, Option or Bundle product. For example, if the maintenance fee of an Option product is 10% of the over all maintenace fee of the main Bundle product, you must create two separate Price List Items for both the products. The Price List Item of the Option product should contain the related pricing fields populated with the values from the Bundle product.

You must perform the necessary configurations at the Price List Item to enable related pricing.

- Add related pricing information on a Price List Item
- Create a record in Related Price List Item (To) related list

Adding related pricing information on a Price List Item

In a Price List Item record (Option product in our example), after you select **Price Method** = *Related Price*, the different fields used for Related Pricing, such as **Related Adjustment Amount**, **Related Adjustment Type** and **Related Adjustment Applies To** are displayed.

Field	Description
Related Adjustment Amount	Indicates the adjustment amount that you want to apply on the related product. In our example, the value should be 10.
Related Adjustment Type	Indicates the type of adjustment that you want to apply on the related product. Available values are: *Percentage*, *%Discount*, *Discount Amount*, *%Markup*, and *Markup Amount*. In our example, choose *Percentage*.
Related Adjustment Applies To	Indicates on which price the adjustments for the related price should be applied. Available values are: *Base Price*, *Base Extended Price*, *List Price*, *Extended Price*, *Net Price*, and *Net Unit Price*. In our example, choose *Base Price*.

Creating a record in Related Price List Item related list

In order to relate two Price List Items and implement related pricing, you must create a record in the Related Price List Item (To) record.

Navigate to **Related Price List Item (To)** related list and click **New Related Price List Item**.

Enter the requisite information:

Field	Description
Price List Item	Denotes the ID of the Price List Item for which you are creating this record.
Related Price List Item	Denotes the ID of the Price List Item whose price is used as a basis for the related pricing.
Relation Type	Denotes how the Price List Items are related with each other. Currently, *Related Price* is supported.
Related Adjustment Amount	Indicates the adjustment amount that you want to apply on the related product. In our example, the value should be 10. You can specify this amount either here or at the Price List Item level.
Related Product	Indicates a filter that you can consider as a source for related pricing. This field must be clubbed with Related Charge Type for correct functioning.
Related Charge Type	Indicates the Charge Type for the Related Price List Item. This should be validated with the Charge Type on the Price List Item record.
Related Product Family	Indicates a filter that you can consider as a source for related pricing. This field must be clubbed with Related Charge Type for correct functioning.
	Indicates a filter that you can consider as a source for related pricing. This field must be clubbed with Related Charge Type for correct functioning.

Field	Description
Related Product Group	
Related Adjustment Type	Indicates the type of adjustment that you want to apply on the related product. Available values are: Percentage, % Discount, Discount Amount, %Markup, Markup Amount. In our example, choose *Percentage*.

Setting the context for Related Pricing

You can set the related pricing context for Line Items using a new custom setting, **Related Price Scope**, available at **Setup > Build > Develop > Custom Settings > Config System Properties > Manage**. This setting enables you to choose if you want to perform the related pricing calculations over the entire Cart or confine the calculations to a Bundle product only. For example, if the price of an Option product is a percentage of another Option product in the same bundle, consider the following scenarios with **Related Price Scope** = *Cart* and *Bundle*:

- **Related Price Scope** = *Cart* , the Option price is based on the price of all the instances of the related product in the Cart, whether it is in the same Bundle or a Standalone or in an another Bundle.
- **Related Price Scope** = *Bundle* , the Option price is based on the related product instances within the same Bundle . You must also ensure that the Option products are a part of that bundle.

Enabling Price Ramps

Price Ramps are enhanced pricing methods for service/subscription-based industry.

As an administrator, you must enable price ramps from the price list item details page. Once you enable price ramps, an end user can create a price ramp for a product to spread pricing across time periods. The user can also create user quantity ramps to define different pricing across different quantities.

> You cannot make a price list item rampable and tierable at the same time.

To enable price ramps

You must have an existing price list item.

On the Price List Item page, select **Enable Price Ramp** check box.

The price becomes rampable for a product.
After the price is rampable, you can create and save price ramps from the configuration page and the pricing cart page.

Enabling Tiered Pricing

Price Tiers enable an end user to view or set up tiers on the configuration page and the pricing cart page. This ensures there are pricing settings (product line item) that determine if tiers can be modifiable or just viewable on the configuration page and the pricing cart page.

You can view existing pricing tiers and adjust pricing on the tier lines (if allowed). This also provides the capability to do tiers per units or usage and also capture additional usage charges for tiers.

If you have defined price matrices, the tiered pricing for a product is inherited from there. If you make any adjustments on the line item that has a tiered rate defined, the adjustments apply to tiered rate and also affect the price matrices.

If you are using usage-based pricing and have a matrix defined, the first dimension should always be a numeric value such as Quantity and not a string value.

> **Important**
>
> You cannot make a price list item rampable and tierable at the same time.

To enable tiered pricing

You must have an existing price list item.

1. On the Price List Item page, from **Price Type**, select **Usage**.
2. From **Price Method**, select **Tiered Rate**. If Tiered Rate is not available as a choice, you can add it to the custom field object.

3. To enable a user to modify a usage tier, select **Is Usage Tier Modifiable**.
4. Click **Save**.

Tiered pricing is enabled. If the price list item has tiers defined on the price list already, predefined tiers are displayed on the options/configuration or the pricing cart page.You can create or modify tiers for pricing on the options/configuration and the pricing cart page.

To include a picklist value in an object

1. Navigate to **Setup > Create > Objects**.
2. Click the custom object. For example, click **Price List Item** label next to the Apttus Configuration & Pricing installed package.
3. Scroll down to the Custom Fields & Relationships related list and click the custom field. For example, click the **Price Method** field.
4. Scroll down to the Picklist Values related list and click **New.**
5. Type a picklist value in the text area. For example, *Tiered Rate*.Note: You can add more than one picklist values. Each picklist value must be on a separate line.
6. Click **Save.**

A picklist value is added to an object field.

Running a Criteria Maintenance Job

Criteria maintenance is a batch job that must be run whenever changes are made to pricing criteria fields or constraint criteria fields.This includes:

- Adding or removing criteria in pricing fields.
- Any modification to price list, price list item, price matrix, price rule, default rule, or price dimension.
- Adding or removing a constraint rule.
- Adding or removing criteria in constraint rules.
- Any change to the constraint criteria fields.

To run a criteria maintenance job

1. Navigate to the Criteria Maintenance tab.

2. To update all pricing criteria fields, click **Update Pricing Fields**. - OR - To update all constraint criteria fields, click **Update Constraint Fields**.

This executes an asynchronous batch job that maintains the criteria changes. The administration task is complete and an updated history for all the batch jobs is displayed. The key item to observe is Status. When Completed is displayed it means the job has run successfully, even if the percentage indicator remains at 0%.

Contract Pricing

Contract Pricing in CPQ enables you to store contracted prices in the system. When a quote is created in CPQ, the quote price is determined by looking up the price defined in the Contract. Contract price list is a subset of the standard price list. This implies that when the same product is defined in multiple contract price lists, the former contract takes precedence. Contract Pricing is supported for standalone and bundles. Contract Pricing for Options is supported provided options are defined in the same contract price list as the bundle. Options cannot be in a different price list. Options inherit the contract price list from the bundle.

Lets understand this with a scenario: Foregen is a company that manages their contracts outside of Apttus. They have around 40-50 contracts. The contracts can be with an individual customer, health network, or a GPO (purchasing organization). A given customer can be part of the health network and/or a GPO. Therefore, they can have multiple contracts that can apply to them. When a customer creates a quote, depending on the product they are quoting for, there is a pre-defined sequence of contract lookup. For example, for product A, the sequence is Individual contract, GPO Contract, Health Network Contract, but for product B, the sequence can be GPO, Health Network, Individual Contract.

Contracts are managed outside of Apttus and they are represented as Price Lists in CPQ i.e for each contract, a price list is created in CPQ and is associated to that Contract (with an identifier on the Price List). When the contract price is changed, the customer has to manually update the contract number in the Price List. On the Account object, you must create 3 custom fields to keep the list of contract that are applicable to the customer and is maintained by the users (manually). For example: There is a field "health network contract", which identifies the contract number that this customer has, as "HEA1234". Similarly, other fields identify the other types of contracts that this customer can buy off of such as GPO "GPO232", Customer "Cust5992". At the time of quoting, a special custom logic (implemented via Salesforce workflow rules), figures out the list of contracts and their sequence and stores it in a field on the line item. For example, it can identify 3 contracts, (HEA1234, GPO232, Cust5992) and will store them as comma separated values.

When looking up prices, the system first looks up for the price from the price list that is associated with the first contract (i.e HEA1234). If the price is not found in that price list, the system will look at the second contract, and if not found in the third. If the price is not found in any of these price lists, the price is retrieved from the standard price list (not associated to the contract).

To enable contract pricing in CPQ, customers need to create a price list to store the contract prices and associate the price list a Contract number. In the quoting module, they need to define (explicitly) which contracts to use for a certain quote or quote line item.

- Associating a Contract Number to a Price List
  To enable contract pricing in Apttus CPQ, you need to create a price list to store the contract prices and associate the price list a *Contract number*.
- Defining a Contract to be used in a Quote
  In the quoting module, you need to define (explicitly) which contracts to use for a certain quote or quote line item.

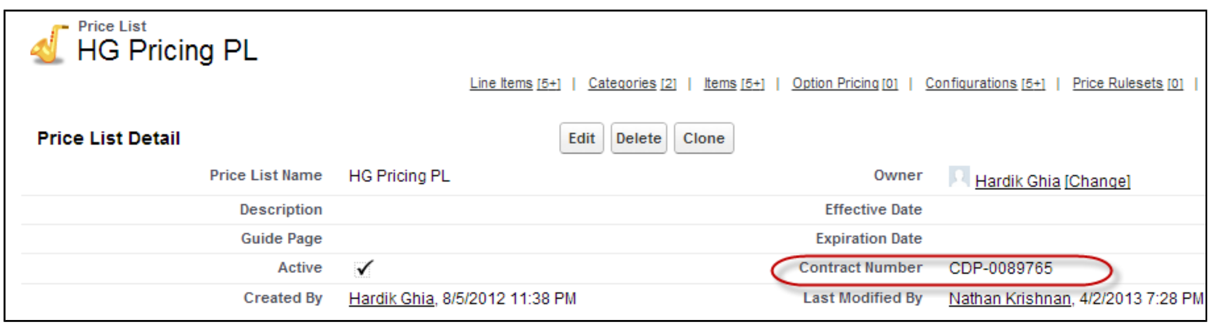
Associating a Contract Number to a Price List

To enable contract pricing in Apttus CPQ, you need to create a price list to store the contract prices and associate the price list a Contract number.

To associate a contract number to a price list

You must have an existing price list. You must have an existing contract number or agreement number.

1. Navigate to the **Price List** tab and select a price list.
2. Click Edit.
3. In **Contract Number**, type a contract number or an agreement number and click **Save**.



A Contract Number is associated with a price list.

Defining a Contract to be used in a Quote

In the quoting module, you need to define (explicitly) which contracts to use for a certain quote or quote line item.

You can define the contracts to be used in a quote at:

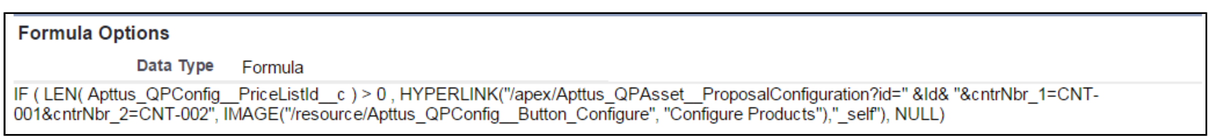
- Quote Level: Use a specific contract or a set of contracts to explicitly pass the contract number(s) in the Configure Products action.
- Quote Line Item Level: Use a specific contract (or a set of contracts) to store the contract numbers separated by ampersand (&) in a field on the quote line.

When the quote is priced, for each line item, first the price list associated with the first contract on the list is evaluated. If the part is not found on the price list, then the next contract number is evaluated, and so on. If part is not found in any of the price lists associated with the contracts on the list, then the standard price list is used.

To define a contract to be used in quote

You must be in the **Quote/Proposal** object.

1. Go to **Setup > App Setup > Create > Objects**.
2. Scroll down the list and click **Quote/Proposal**.
3. From **Custom Fields & Relationships**, click **New**.
4. From **Data Type**, select **Formula** and click **Next**.
5. Enter a mandatory **Field Label** and press **Tab** on your keyboard, the **Field Name** is auto-populated. For example: In Field Label, enter **Configure Products (Contract Pricing)**, the Field Name is auto-populated with **Configure_Products_Contract_Pricing**.
6. From **Formula Return Type**, select **Text** and click **Next**. The **Step 3. Enter formula** page is displayed.
7. Type the code in the area displayed and click **Next**. See example code.


Formula Options

Data Type Formula

```
IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 , HYPERLINK("/apex/Apttus_QPAsset__ProposalConfiguration?id=" &Id& "&cntrNbr_1=CNT-001&cntrNbr_2=CNT-002", IMAGE("/resource/Apttus_QPConfig__Button_Configure", "Configure Products"),"_self"), NULL)
```

8. On the **Step 4. Establish field-level security** page, make the necessary changes and click **Next**.

9. On the **Step 5. Add to page layouts** page, select the page layout that should include this field and click **Save**.

The Configure Products (Contract Pricing) action button is displayed on the Quote/Proposal page.

Alternately, you can also create a field on the Quote header and enter comma separated contract numbers in that field. Use the following formula text for the Configure Products button. The system automatically looks into contract numbers provided in sequence.

```
IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 , HYPERLINK("/apex
/Apttus_QPConfig__ProposalConfiguration?id=" &Id& "&cntrNbr_1="&Contr
act_price_list_1__c,IMAGE ("/resource
/Apttus_QPConfig__Button_Configure", "Configure Products"),"_self"),
NULL)
```

Pricing Profiles and Batching Price Calculations

Pricing profiles and batching price calculations enable you to specify the type of pricing configured so that CPQ can optimize run time price calculations.The following two custom settings control this behavior:

- **Pricing Profile**
- **Pricing Batch Size**

You can specify the **Pricing Profile** as *Basic* or *Advanced*.By definition, a pricing profile is *Basic*, if:

- There are no pricing rules used
- There are no price matrices used
- There are no related pricing setup
- There are no bundles

At present, Apttus supports only two pricing profiles: *Basic* and *Advanced*. If the Pricing Profile field is left blank, the default value is *Advanced*. If the Pricing Profile is *Basic* and Defer Pricing check box is selected, the performance can be improved by executing price calculations in batches. To enable and control batching of price calculations, use the Pricing Batch Size field.

The Pricing Batch Size setting defines the number of line items that can be processed in a single pricing call. Setting the Pricing Batch Size, the system runs pricing with the specified number of products as a batch, thus increasing performance. These batch calls to the database is governed by Salesforce CPU time limit and hence the number assigned for Pricing Batch Size has to be carefully evaluated.

> These two system properties are global and affect the entire CPQ implementation in an Org. You must ensure that all the products have only Basic pricing configured as defined above.

To set up pricing profile and pricing batch size

1. Go to **Setup > Develop > Custom Settings**.
2. Click **Manage** for **Config System Properties**, and click **Edit.**
3. Select **Defer Pricing**.Note: To use **PricingProfile** and **PricingBatchSize**, **DeferPricing** must always be selected.
4. For **Pricing Profile**, type *Basic* or *Advanced*.Note: If this field is left blank, the default value is *Advanced*.
5. For **Pricing Batch Size**, type a value.

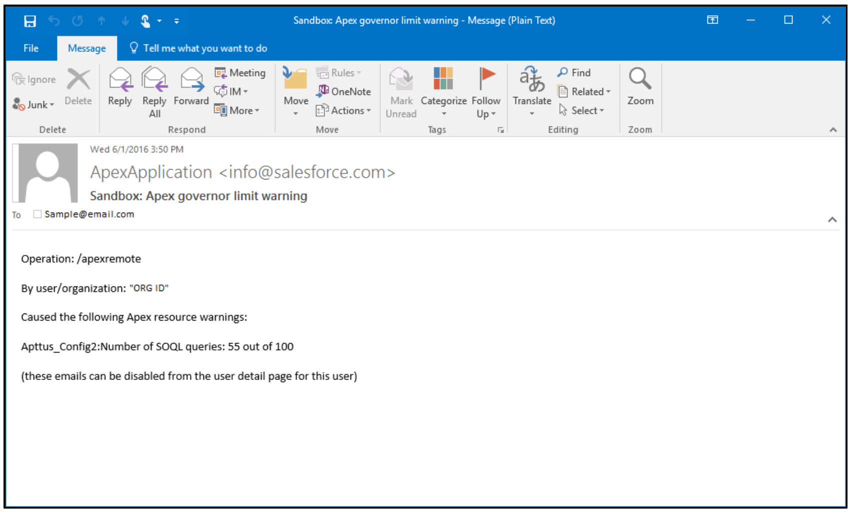
   > If this field is left blank, the default value is 1.

6. Click **Save.**

Pricing is performed as defined in Pricing Profiles and batches as defined in Pricing Batch Size.

Apex Governor Limit Warning Email Notification

When Apttus CPQ executes any action (such as pricing or configuration), your users may receive an email notification indicating that they have reached a certain lower threshold for the Apex governor limit. This is a normal behavior and Apttus CPQ is built with these limitations in mind with a certain degree of buffer for further actions. As an Admin, you can disable the email notifications from the user detail page for a specific user.



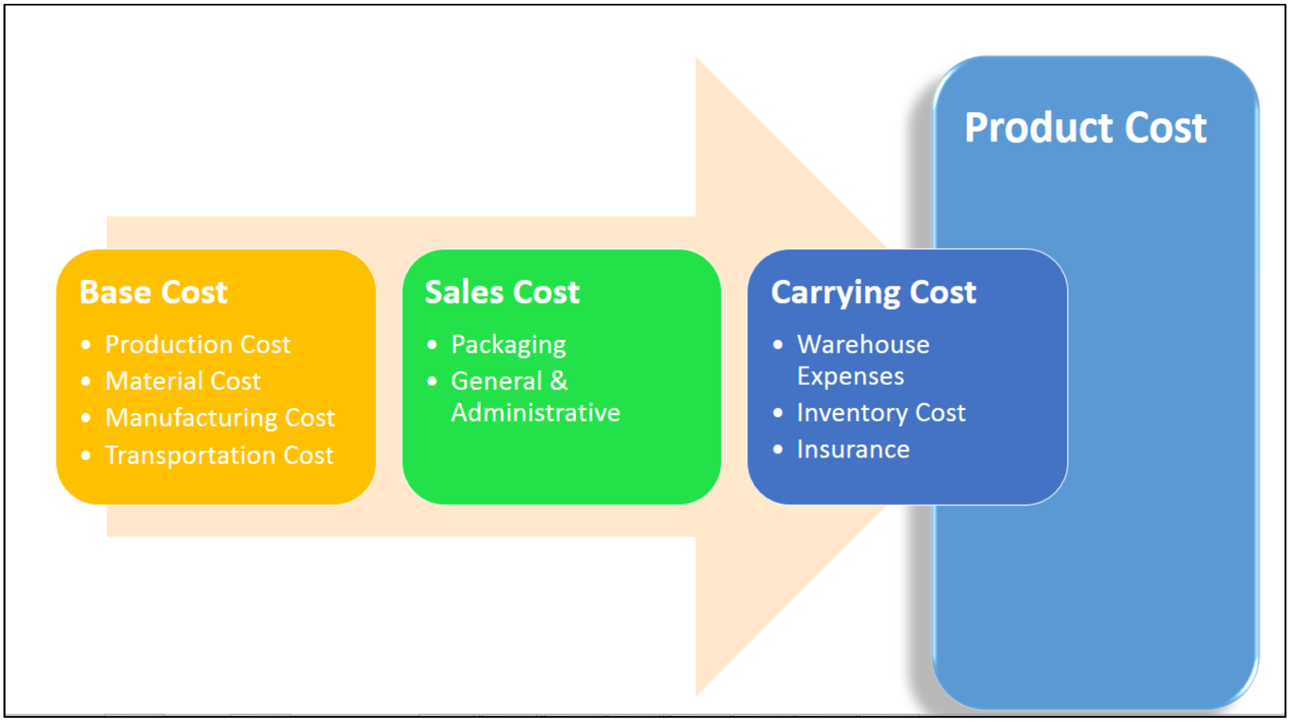
Cost and Profitability (Cost Breakdown)

Cost of a product, as you see it, is derived from aggregating direct, indirect and variable costs. These costs could be manufacturing, Sales, Administrative, volume-based or any other miscellaneous cost.
For example, the Cost of a Drilling Machine is an amalgamation of costs from various sources incurred during the life cycle of the product - such as raw material, production, labor, time, taxes and other marginal costs. While some of these costs remain stable, some vary depending on the volume, availability or other market conditions.

To cite another example, the price of a watch is derived from the summation of the cost of designing, manufacturing, packaging and shipping. Similarly, if your product is a service, cost can be derived from several dependent factors such as infrastructure, security, fulfillment and others. In a nutshell, for all products, such finer cost elements that are associated with the product life-cycle, together constitute the final cost.

As a Sales Rep, when you are adding products to the cart during the Quote generation process, adding margins and reaching the profitability of the deal will become easy if you can see the cost elements which is precisely the cost break-down of a product.
Visibility of the cost hierarchy for a product that captures every granular cost will empower you to prepare a profitable Quote that leaves no room for generalization and approximation.

When you know the key cost drivers and can see how what other costs contribute to the key cost, you know which costs should be tweaked or modified for arriving at an optimum cost.



The cost models, as you can see in the above illustration, are nothing but the structure of cost elements and sub-elements participating in deriving the final cost. The cost hierarchy gives you the granular level of details regarding the various aspects of product costing.
With Apttus CPQ, you can create complex Cost Models that give insights into the true profitability of a product price. During the quotation process, this transparency of costs can provide accurate deal margin decisions that are critical for thriving in the competitive market.

Start implementing the cost model for your products to

- Achieve Effective Deal Guidance,
- Eliminate the need for an external Cost Management application, and
- See the price breakdown and assess the profitability of your quote.

Key Terms

Term	Description
Base Price	The basic product price without break-ups.
Mark-up	Additional cost incurred for a product to create a profit.
Fixed Costs	Costs that do not vary for a long period of time are considered as Fixed Costs. For example, Maintenance Costs, Renting or leasing of assets, Property Insurance etc.
Variable Costs	Costs that are volume-driven or change frequently with time are regarded as being variable. For example, raw materials, labor, social expenses, training etc.

There are three key facets of this feature:

- **Price breakdown**
  Earlier you could associate only a single cost to the product. Because the cost is dependent on various factors some of which might vary, a single field defining the cost was very limiting. With cost models in place, you can define the variable and fixed cost elements and achieve dynamic product costing. This approach is the best way to ascertain what a product should be priced at.
  To see how you can configure and store the granular costs associated with your product, please refer Storing Price of Products.
- **Transfer Price**
  Setting the price for goods or a product between legal entities within an enterprise is known as Transfer Pricing.
  The transfer price is the price at which different goods and services are transferred between divisions within an organization. For example, a division of Motor Company- A, based in South Africa supplies Gears to their global assembly plant in Boston. The division in South Africa will charge the division in South Africa with the cost of the Gear including their profit margin. This price should be an amount that benefits the organization and also allows each department to have a fair share of their profits.

Majorly, a Business Unit or Factory "sells" to the front end sales team. There are regulatory requirements around how the transfer price is calculated. OECD (Organization of Economic Cooperation and Development) has set some guidelines and most countries follow or add-on to those guidelines. To know how you can create a Transfer Price mapping, please refer Transfer Price.

- **Price Waterfall**
  You can Analyze a Quote by looking at the Price Waterfall chart and Waterfall table for corresponding Line Items and Summary groups. The waterfall chart is a bar graph where Line Item Fields are plotted vertically and Cost is plotted horizontally. For details, please refer Analyse Quote.

We'll take a detailed look at each of these aspects in the later sections.

Required Configurations

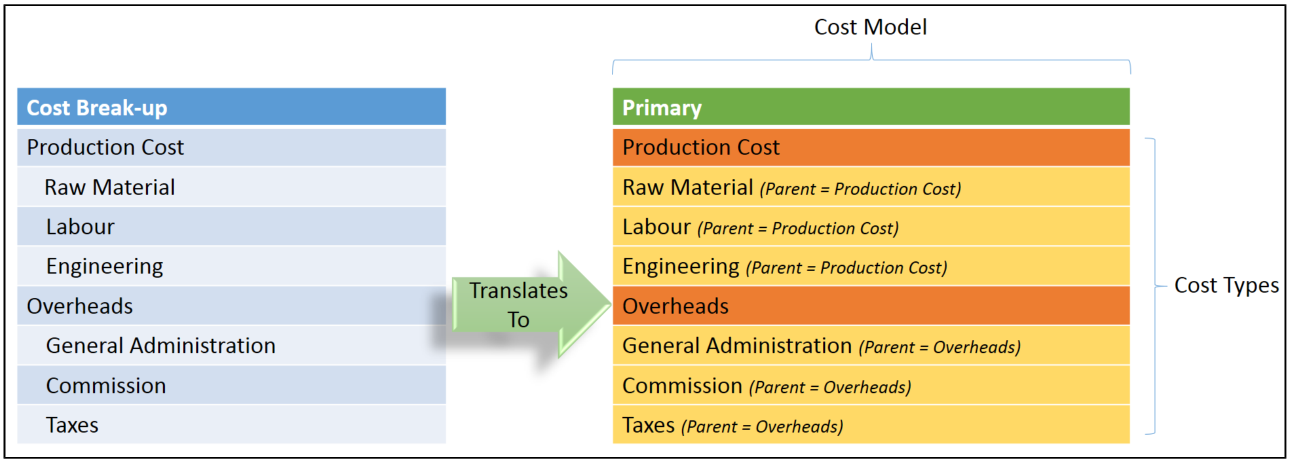
A way to arrive at accurate costing is to dissect the cost into different elements, roll-up this cost and apply markups to finalize it. These elements are identified as Cost Types in the system.

Let us see how you can start creating a cost hierarchy for your products and analyze the margins on the cart. To get started, you need to:

1. Define a **Cost Model**
   a. Set a Cost Hierarchy by adding **Cost Types**
2. Associate the Cost Model with a Price List.
3. Configure a Presto App (using X-Author) to store product prices.
4. Fetch prices for the Cost Types by adding a Presto Callback
5. Analyze pricing from **Analyze Quote** button on the cart.

Defining a Cost Model

Cost Model is a container holding all Cost Types. As shown in the diagram, you can create *n* Cost Types for *n* number of Cost elements.



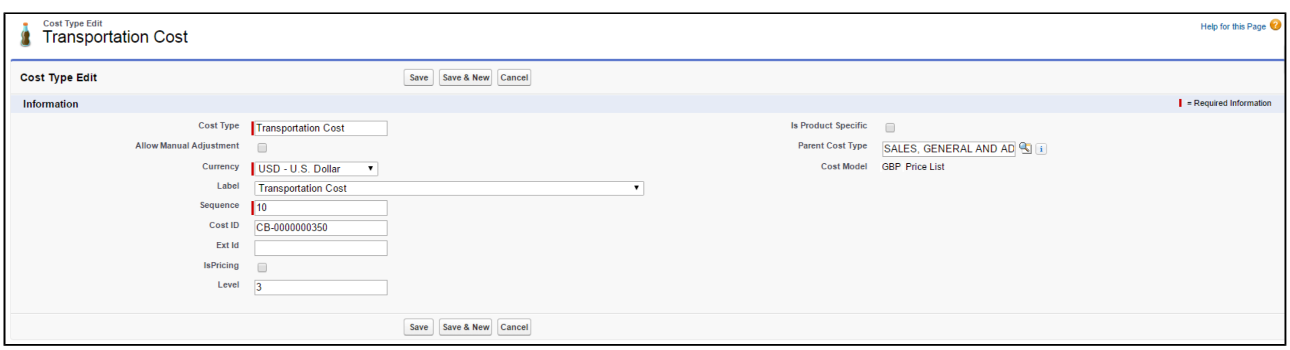
1. Go to **+** (**All Tabs**) > search for **Cost Model** and click **New**.

2. Enter a *name* for the Cost Model, choose a **Currency** and **Save** your changes.

From the Cost Types Related List, start adding cost types by clicking *New* **Cost Type**.

Define Cost Types

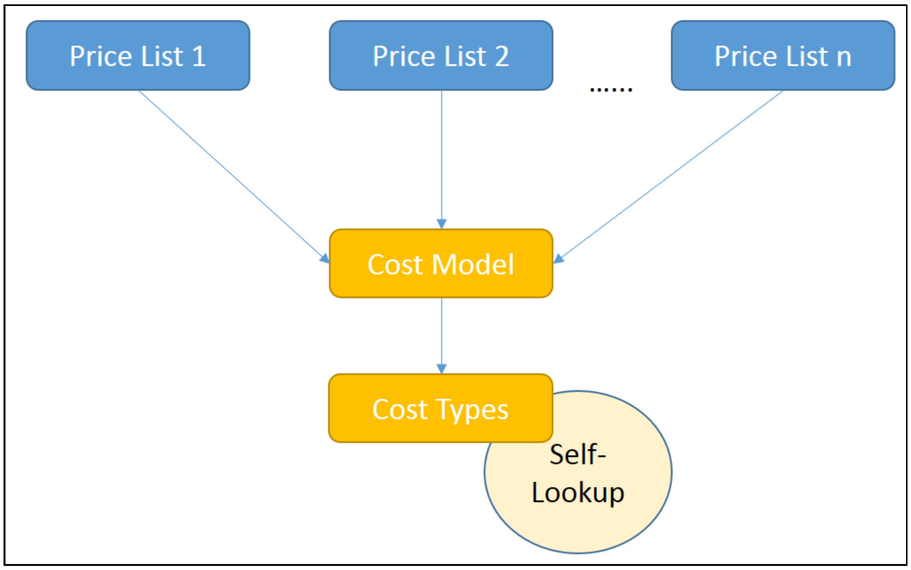
Cost Types correspond to individual cost entities. You can define *parent* for a Cost Type thereby creating Cost Type hierarchies.



Option	Description
Cost Type	The name for a Cost Type.
Currency	The Currency specific to this Cost Type.
Label	A Label differentiating the Cost Type.
Sequence	The order for arranging hierarchies sequentially.
Parent Cost Type	Choose a Parent Cost Type for the said Cost Type.
Cost Model	Read-only field hat indicates the Cost Model associated to this Cost Type.
Allow Manual Adjustment	If checked, allows you to override costs on the **Analyze Quote** page. On clicking Reprice, the base price is recalculated with the new costs.

Associating Cost Model with a Price List

On the Price List detail page, look for the Cost Model field and apply a Cost Model of your choice. You can associate one Cost Model with multiple Price Lists. However, one Price List can have only a single Cost Model associated with it.

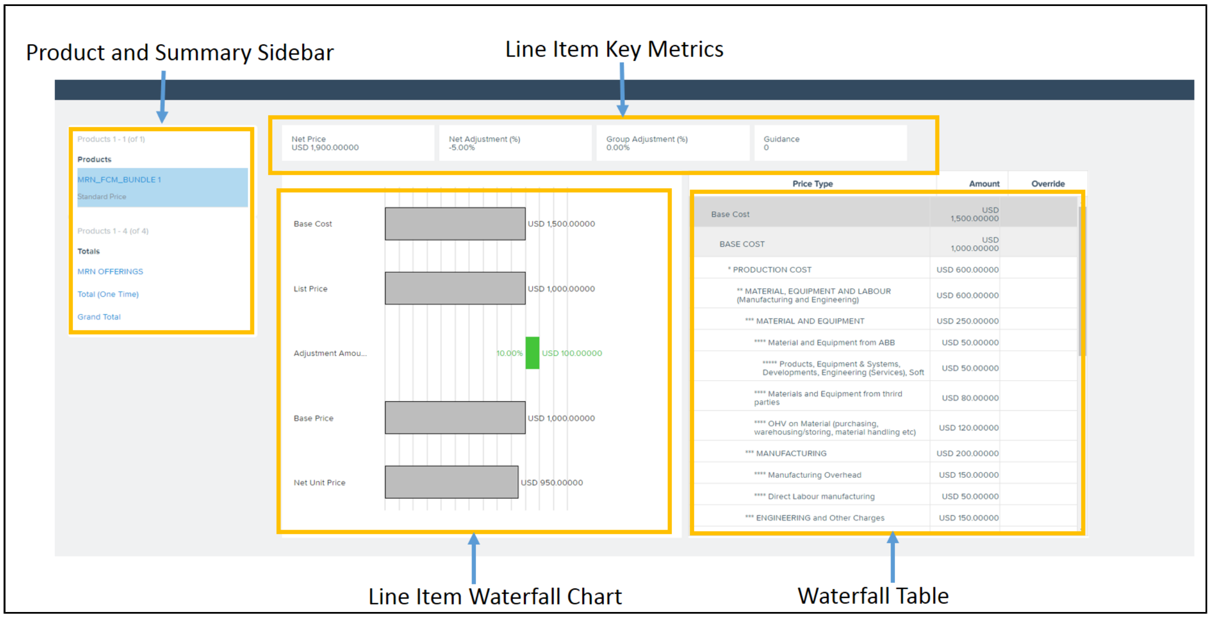


Configuring Analyze Quote Page

Click **Analyze Quote** button on the cart, to load the **Analyze Quote** page.

Analyze Quote page is divided into following four components:

1. Product and Summary Sidebar
2. Line Item Key Metrics
3. Line Item Waterfall Chart
4. Waterfall Table



You can opt to display the fields of your choice in the **Line Item Key Metrics** and **Line Item Waterfall Chart** screen elements for both **Line Items** and **Summary Group options**.

Go to **Config Settings > Display Column Settings** and select **Display Type** as *Line Item Key Metrics* for configuring fields displayed on top of the page.

Similarly, you can configure the display for *Line Item Waterfall*, *Summary Group Key Metrics, Summary Group Waterfall* and *Optional Cost and Profitability Fields.*

All the fields of Cost Line Item, Adjustment Line Item and Cost Adjustments are available to be displayed on the Analyze Quote page.

Storing Price of Products

The Price for each Cost Type is defined in an Excel App. You have to replicate the Cost Model, defining various Cost Types and their pricing with the help of X-Author Designer. When you click Configure button on the Quote/Proposal, pricing data from this app is fetched and shown for analysis.

To get started, you will need to Install **X-Author** and **Designer** and **Runtime** packages.

App Setup

Install the following packages:

- X-Author for Excel (3.25 or higher)

- Apttus X-Author™ For Excel Document Generation (1.0)
- Apttus Document Generation (1.5)
- X-Author Designer for Excel

Getting Started

1. In your Salesforce org, create a **Cost Model** and add Cost Types.
2. Open X-Author Designer and create a pricing app.
   For information on how to create an app, please refer- Creating Apps with Quick Apps
3. Ensure the Cost Type Label and Cost ID matches with the Cost Names specified in the org.
4. On the App, add Parent Code (from the Salesforce URL) for the child Cost Items.
5. **Validate** and **Save** the App.
   Go to your Salesforce org and access Apps tab to see your app.
6. Edit the app and select **Activated** to activate it.
7. Create a **Price List** and associate it with the Cost Model.
8. In order to run this App while configuring products, pass the App Id in the 'prestoAppId' field in the following formula.

```
"IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0, HYPERLINK("/ap
ex/Apttus_QPConfig__ProposalConfiguration?id=" &Id
+"&flow=NGDefault"&"&useAdvancedApproval=true&useDealOptimizer=t
rue"&"&prestoAppId=a5t2C000000GnBe",
IMAGE("/resource/Apttus_QPConfig__Button_Configure", "Configure
Products"),"_self"), NULL)"
```

9. Go to **+** (All Tabs) > **Apps**, open your app and copy the App Id from the URL.
10. Create a Quote, add the product to the cart and click Reprice. To see the price breakdown Click **Analyze Quote** button.

Transfer Pricing

Transfer Pricing is the price at which two related parties perform business transactions between themselves. Every organization follows a regulatory guideline specific to their country to derive transfer prices for the products. This formula states one of the ways in which Transfer Pricing is calculated.

Transfer Pricing = **Full Cost** (sum of all fixed and variable costs) + **Markups**

Let's assume the Sales Rep exposes a currency field 'Manufacturing Cost' and sends collaboration request to another Factory user.

Cost Model available with the Sales Rep

Cost Name	Value
Packaging	$100
Transportation	$50
Manufacturing	-

Cost Model with the Factory User

Cost Name	Value
Manufacturing	$70 (Cost of Raw Material + Labor)
Raw Material	$30
Labor	$40

The administrator needs to create a Transfer Map so that Sales Reps can send out a collaboration request. For this requirement, the Transfer Price Map would look as the following:

Field	Value
Source Field	Apttus_Config2__Manufacturing__c
Source Charge Type	Standard Price
Target Cost Model	Factory Cost Model

Field	Value
Target Field	Requested_ManufacturingNetPrice__c
Target Object	Apttus_Config2__LineItem__c

Configuration

From **All Tabs** (+) go to **Transfer Prices** and Click **New** to create a Transfer Price mapping.

Field	Description
Source Field	The *API Name* of the field for which you want the pricing details.
Source Charge Type	Mention the Charge Type of the Source Field, whether it is a Standard Price, License Fee, Subscription Fee or any other.
Target Cost Model	Enter the name of the destination Cost Model from which you want to get the price of a field.
Target Field	Enter the *API Name* of the field as given in the Target Cost Model.
Target Object	Enter the *API Name* of the Target Object which the Target Field is a part of. For example, if the Target Field is 'Net Unit Price' and is defined on the Line Item object, the Target Object will be Apttus_Config__LineItem__c.

Procedure

1. Create a new Quote using price list.
2. Click Configure Products button. Add products and go to Cart.
3. Using the collaborator icon send the request to Factory User.

4. Factory User receives the Collaboration Request. Applies adjustments or enters the price and clicks **Submit For Merge**. Please refer Assigning the configuration to the Collaborator or Queue for details.
5. The Transfer price is set to copy **Net Price** from Collaborator's Quote to a cost type named "Cost From Factory" of requester's Quote.
6. Open the requester's Quote. Go to Cart and click **Merge** in the Collaborator pop-up.
7. Click **Analyze Quote** to analyze the product cost.

Rule Management

Working with Constraint Rules

Constraint Rules are configuration rules that are applied to a product or a set of products during the quoting process. Constraint rules are configured for conditional addition of a product to a quote or conditional blocker of product additions to a quote.

For example, when Product A is added to the cart, Product B may be automatically added based on an auto inclusion type rule. Similarly when Product B is added user may be prevented from adding Product C.

Constraint rule consists of three associated Objects:

1. Contraint Rule Object: This is a header level object that links the condition with the action. There are only two significant properties in this object Active and Condition Association. To configure a constraint rule, you must first create a constraint rule header.
2. Constraint Rule Condition: This object captures the condition that triggers the rule.
3. Constraint Rule Action: This object captures the rule action that is applied to the cart when the rule condition is met.

After creating constraint rule action, on the constraint rule detail page, click the **Update View** button. This runs the constraint rule maintenance for the context rule. You may have to drag and drop the **Update View** button using the Edit Layout feature.

Apttus supports five types of constraint rules:

- Inclusion - Inclusion type rule is used to add a product to the cart
- Exclusion - Exclusion type rule is used to exclude or prevent addition of a product to the cart
- Replacement - Replacement type rule is used to replace and add a product to the cart
- Recommendation - Recommendation type rule is used to prompt product suggestions to the user for addition to the cart
- Validation - Validation type rule is used to prevent the user from finalizing the cart without resolving validation errors

You can also execute a constraint rule across the following bundle structures:

- From parent bundle structure to nested bundle structure
- From next bundle structure to parent bundle structure
- Across nested bundles

Use case: For a bundle product X, you have an option group OG 1. There is also a sub-bundle, with Option Group 1.1, containing options, Option 1, Option 2 and Option 3. Bundle X also contains an Option Group OG2, with options, Option A, Option B, Option C.

Bundle X

-----OG 1

-------------- Sub-bundle 1

----------------------OG 1.1

------------------------ Option 1

------------------------ Option 2

------------------------ Option 3

-----OG 2

-------------- Option A

-------------- Option B

-------------- Option C

You can now write a rule to auto include Option A when a user manually selects Option 1.

To create constraint rules

1. Click and click **CPQ Console**, and from the Rule Management section, click **All Constraint Rules**.
2. Click **New Constraint Rule** and type a mandatory **Constraint Rule Name**.
3. Type a **Description** and **Sequence**, and to make the constraint rule active, select **Active.** The constraint rule applied to the configuration only when it is active.
4. Enter an **Effective Date** and **Expiration Date**. These fields are for capturing whether the rule is effective at the time of the configuration process.

5. In **Condition Association**, define additional criteria to further filter the product scope condition. When there are more than one condition for a constraint rule, this field provides when to consider one or more of them using AND/OR association. The sequence number in the constraint rule condition is used as identifier of the condition.
   Few examples:
   a. (1 AND (2 OR 3))
   b. 1 AND NOT 2 — 1 is in the cart while 2 is not
   c. 1 AND NOT(2) — 1 is in the cart while 2 is not
   d. NOT(5 OR 6) — neither 5 or 6 are in the cart
   e. NOT 5 AND 6 — 5 is not in the cart, but 6 is in the Cart
6. Click **Save.**

> You must set the Sequence for each condition or the Condition Association will be unreliable. Also, a constraint rule with only NOT in condition will not work. Example, NOT (1 AND 2).

The constraint rule header is created.

You can now create a constraint rule condition.

- Creating Constraint Rule Conditions
  Constraint rule conditions should be met in order to trigger a rule. There may be more than one condition for a rule. Different conditions are associated with an expression. When the condition association expression is not specified, all the conditions should be met in order to trigger the rule.
- Creating Constraint Rule Actions
  Constraint rule actions are performed, when certain constraint rule conditions are met. A rule action can be of inclusion, exclusion, validation, recommendation, or replacement type. An inclusion rule action enforces selection of certain products as specified by the product scope specification or criteria specification in the inclusion rule. Similarly an exclusion rule action enforces exclusion of certain products from the shopping cart. The validation rule displays an error message, when the criteria conditions are met. The recommendation rule displays products based on the condition while the replacement rule displays message with products that can be replaced. The recommendation rules support product, product family and product group.

- Translating Custom Messages For Constraint Rules
  Custom messages for Constraint Rules can be translated in multiple languages based on the User's org language.
- Constraint Rule Maintenance
  Constraint Rule maintenance is a batch job that must be run whenever changes are made to rule conditions and when you upgrade to a new version of CPQ.
- Setting up an Inclusion Rule
  Inclusion rule use case: You have a product A and product B. You want to set up an inclusion rule such that when a user selects Product A, a warning message is displayed to include Product B.
- Setting up an Exclusion Rule
  Exclusion rule use case: You have a product A and product B. You want to set up an exclusion rule such that when a user selects Product A, a error message is displayed to exclude Product B.
- Setting up a Validation Rule
  Validation rule use case: You have a product A and you want to set up a validation rule such that when a user selects Product A and should not be able to enter more than 5 quantity.

Creating Constraint Rule Conditions

Constraint rule conditions should be met in order to trigger a rule. There may be more than one condition for a rule. Different conditions are associated with an expression. When the condition association expression is not specified, all the conditions should be met in order to trigger the rule.

The rule condition is defined by product scope. The product scope could be specific to a Product, Product Family, Product Group, and a Product Field. The additional criteria on the condition is based on the line item object. When a product is in the product scope of a condition, the condition is met when the product is in the cart. When the condition is met, the rule action is applied.

After you create a constraint rule condition, you can define additional condition criteria. Condition Criteria filter conditions are applied to the line item created from the selected products that are within the product scope definition. When criteria is defined using the Edit Criteria button, the condition will be met when the product scope specification and criteria are both met. Condition Criteria filter conditions are applied to the line item created from the selected products that are within the product scope definition.

When using a constraint rule with a condition criteria for a product with multiple line items, the system only validates the primary line item to evaluate the constraint rule.

You must run a criteria maintenance job whenever you add, remove or make any changes to a criteria. Even if you remove a constraint rule that has a criteria defined, you must run a criteria maintenance.

To create constraint rule conditions

You must have an existing constraint rule header.

1. Select an existing constraint rule header and scroll down to the Constraint Rule Conditions related list, and click **New Constraint Rule Condition**.
2. Type a mandatory **Constraint Rule Condition Name**, verify the **Constraint Rule** and type a **Sequence**.
3. In the Criteria section, from Product Scope, select one of the following:

Option	Description
Product	After selecting, enter a product in the Product field. This implies that the product should be present in the shopping cart in order to meet the condition.
Product Family	After selecting, enter a Product Family in the Product Family field. When specified, one or more products from the specified Product Family should be present in the shopping cart in order to meet the condition. The number of products that should be present depends on the Match Rule.
Product Group	After selecting, enter a product group in the Product Group field. This implies that one or more products from the specified product group should be present in the shopping cart. The number of products that should be present depends on the Match Rule.
Product Field Set	After selecting Product Field Set as the product scope, select at least one Product Field and set the corresponding Product Field Value. Each of the Product Field has one pick list value from the product object.Note: The Product Field should be set up to have only one choice. If the choices are changed at a later time, the constraint rule conditions with Product Field Set specification should also be updated to have the correct values.Match Rule is not applied to Product Field Set based specification. It assumes the match rule as match any from the result.

4. In Match Conditions section, do one or more of the following:

Option	Description
Match In Options	Selecting this means the condition is met only when the condition products are in the option line items of the shopping cart.
Match In Primary Lines	Selecting this means the condition is met only when the condition products are among the primary lines (bundle or standalone) of the shopping cart.
Match in Cart Options	Selecting this means the condition is met only when the condition products are among the cart line items of the shopping cart.
Match in Location	Selecting this means that when the condition is met, all of the rule actions should be applied within a location.
Match Rule	Select one of the following choices. The value is None by default. Match count is done before applying the criteria. • None means that the rule condition is met, if any one of the products in the product scope is selected. It is treated as Min/Max with Match Min Products as 1. • Match Any In Group means that the rule condition is met, if any one or more of the products from the group is selected. Match Any In Group is same as Include Min/Max with Match Min Products as 1. It is also applicable to Product Family and product field set. • Match All In Group means that all of the products from the group must be present in the shopping cart to meet the condition. This is applicable only to the product group and Product Family. • Min/Max also requires you to specify a value in the Match Min Products field.

Option	Description
Match Min Products	This field is applicable only when you select Min/Max from the Match Rule picklist. The condition is met, when the minimum number of products specified is selected from the Product Group, Product Family, or as specified by the Product Field set specification.

Match Rule is applicable to product scope of type Product Group, Product Family, and Product Field Set only.

5. Click **Save**.
6. To define a condition criteria, scroll down to the Constraint Rule Conditions related list, and click **Edit Criteria**.
7. Select a **Field**, an **Operator**, and type a **Value**. You can enter more than one criteria.
8. Click **OK**.

The constraint rule condition is saved and added.

You can now define a constraint rule action.

Creating Constraint Rule Actions

Constraint rule actions are performed, when certain constraint rule conditions are met. A rule action can be of inclusion, exclusion, validation, recommendation, or replacement type. An inclusion rule action enforces selection of certain products as specified by the product scope specification or criteria specification in the inclusion rule. Similarly an exclusion rule action enforces exclusion of certain products from the shopping cart. The validation rule displays an error message, when the criteria conditions are met. The recommendation rule displays products based on the condition while the replacement rule displays message with products that can be replaced. The recommendation rules support Product, Product Family, and Product Group.

While creating a constraint rule action, when you select product scope as product, you must enter details only in the Product field. Similarly, when you select product scope as Product Family, or Product Group, you must enter details in their corresponding fields only. When you select Action Criteria as the product scope, you must first create the constraint rule action and then edit the same constraint rule action to define an action criteria.

You should be careful not to put too many actions into a single constraint rule. Typically a ratio of no more than 5 actions per 1 constraint rule can help ensure the rules do not impact the performance of your system. You should test the constraint rules thoroughly before pushing them to a live environment.

Action criteria can be set up only when the product scope is Action Criteria. The criteria is defined using the Edit Criteria button. It is used to search for products, when it is defined for inclusion type rule actions. For exclusion type rule actions, the criteria is used to match products in the shopping cart. The criteria can have only the product fields in case of inclusion and exclusion type rule actions.

In case of validation type rule actions the criteria entries are based on the line item fields. Related fields of the line item object, such as the attribute value field can also be used. For validation type rule actions, when the criteria conditions are met, the rule action message is displayed. If the rule action is error type, the end user cannot finalize the configuration until the criteria condition values have changed to make the validation ineffective.

When using a constraint rule with an action criteria for a product with multiple line items, the system only validates the primary line item to evaluate the constraint rule.

You must run a criteria maintenance job whenever you add, remove or make any changes to a criteria. Even if you remove a constraint rule that has a criteria defined, you must run a criteria maintenance.

To create a Constraint Rule Action

You must have an existing constraint rule header with an existing rule condition. Select the constraint rule header. The Constraint Rule detail page appears.

1. From the Constraint Rule Actions related list, click New Constraint Rule Action.
2. In Constraint Rule Action Name, enter a mandatory name for the action.
3. In Constraint Rule, the constraint rule header is displayed for which you are creating an action. You can also click to search for a constraint rule header.
4. In Message, enter text to be displayed when the rule action is effective. These messages can also be translated in multiple languages. For inclusion and exclusion type rule actions product name substitution parameter {0}, {1} can be used. For example: **{0} requires {1}**. For validation type rule actions only {0} will be substituted by triggering products.
5. In Sequence, enter a number to indicate the order in which constraint rule action messages should be displayed.

6. From **Action Type**, select one of the following:

Option	Description
Inclusion	Enforces selection of certain products as specified by the product scope specification or criteria specification in the inclusion rule.
Exclusion	Enforces exclusion of certain products from the shopping cart.
Validation	Displays the errormessage,when the criteria conditions are met.
Recommendation	Product recommendations aredisplayed,when the condition is met.
Replacement	Product replacements are suggested, when the condition is met.

Action Type selection impacts the options available for **Action Intent**.

Constraint rules of **Action Types** = *Recommendation* and *Replacement* do not support **Condition Criteria** and work only for the Product scope.

7. From **Action Intent**, select one of the following:

Option	Description
Auto Include	Adds selected products automatically. When the number of products selected by the rule action is more than the number mentioned in the Match Rule condition,autoinclusionof products is not triggered. A message is displayed from the rule action instead.Note: Auto Include will not apply, when the match rule value is Include All.
Prompt	Displays a message with choices of products that you can select or remove.
Show Message	Displays a message.
	Applies rule action on finalization of the shopping cart.

Option	Description
Check on Finalization	
Disable Selection	Disables selection of excluded products.

8. From **Action Disposition**, select *Error* or *Warning*. The configuration cannot be completed; unless the end user meets all the rule action requirements.
9. In the Criteria section, from Product Scope, select one of the following:

Option	Description
Product	In the Product field, click to select a product to be included or excluded.
Product Family	In the Product Family field, enter a Product Family. One or more of the products from the Product Family will be included or excluded based on rule type and match rule specifications.
Product Group	In the Product Group field, enter a Product Group. One or more of the products from the Product Group will be included or excluded based on rule type and match rule specifications.
Action Criteria	The action criteria is defined using the Edit Criteriabutton,after you save the constraint rule action.

10. In Match Conditions section, do one or more of the following:

Option	Description
Match In Options	Selecting this, the system will prompt you to include or exclude the product as an option based on your Action Type selection.
	Selecting this, the system will prompt you to include or exclude the product as a primary line (offering) based on your Action Type selection.

Option	Description
Match In Primary Lines	
Match in Asset	Selecting this, the system will prompt you to include or exclude the product as an asset line item based on your Action Type selection.
Match in Cart Options	Selecting this, the system will prompt you to include or exclude the product as a cart line item based on your Action Type selection.
Ignore Cart	Selecting this, the system will prompt you to exclude the product as a cart line item based on your Action Type selection.Note: Ignore Cart is applicable only for Exclusion Rule.
Min/Max Match Rule	Applies to inclusion type rule action only, and that has product scope as Product Family, Product Group, or Action Criteria. Select any one of the following to complete the rule action: • None: No product selection is allowed. • Include Any: Allows selection ofupto one product. An error or warning message is displayed upon selection of another product. • Include All Allows selection of all products within the Product Group or Product Family.Note: Only this match rule works with Action Intent Auto Include. • Include Min/Max: Requires minimum number of products specified in the Min Products field. • Exclude Any: All the products are excluded. • Exclude After One: Allows selection of only one product. Any other selection will trigger the rule action. • Exclude After Max: Allows selection ofmaximum number of products specified in the Max Products field. Any other selection over and above the specified number will trigger the rule action.

Option	Description
Max Products	Dependent on Exclude After Max field. Specify the maximum number of products to be selected.

11. Click **Save**.
12. To define an action criteria, scroll down to the Constraint Rule Actions related list, and click **Edit Criteria**.
13. Select a **Field**, an **Operator**, and type a **Value**. You can enter more than one criteria.
14. Click **OK**.

Constraint rule actions are created which will trigger when a constraint rule condition is met.

Translating Custom Messages For Constraint Rules

Custom messages for Constraint Rules can be translated in multiple languages based on the User's org language.

These messages can be translated from the Custom Messages tab. To set up, you require the Message, Message Tag, and the Source Object ID. The Source Object ID is the Constraint Rule Action ID.

You can add as many language as you want that are supported by Salesforce.

To translate custom messages

You must have an existing constraint rule with message.

1. Click and click **Custom Messages**.
2. To create a new custom message, click **New**.
3. Type a mandatory **Custom Message Name**.
4. In Message, type the same message you defined in the constraint rule action. For example: If the Constraint Rule Action Message is *{1} should be suggested when customers select {0}*, the Custom Message must also be *{1} should be suggested when customers select {0}*.
5. Based on the language of the User's profile, type the translated message in the relevant message box. For example: For German, type the message in German language in Message_de box.

6. In Message Tag, type the same message as in the Message box.
7. Type or copy and paste the Source Object Id.
8. Click **Save**.

If the language of the user's profile is German, on the Catalog page, when the Constraint Rule is triggered, the message is displayed in German language.

Constraint Rule Maintenance

Constraint Rule maintenance is a batch job that must be run whenever changes are made to rule conditions and when you upgrade to a new version of CPQ.

This includes:

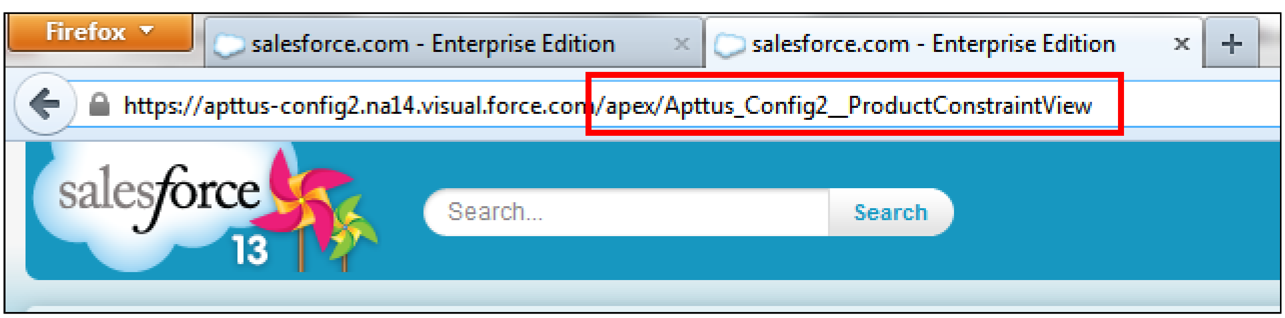
- Adding or removing a constraint rule.
- Adding or removing a constraint rule condition.
- Modifying anything in a constraint rule condition.

Users who are going to run Constraint Rule Maintenance must have read/write access to ProductConstraintView__c object . All CPQ users must have at least read access. These users should also have ProductConstraintView Visualforce page access and Apex class access enabled.

If the Constraint Rule Maintenance tab is not already available, you can add it as follows:

1. Go to **App Setup > Create > Apps** and click **Edit** beside **Apps Apttus Batch Updates**.
2. Add **Constraint Rule Maintenance** and click **Save**. The tab should now be available.

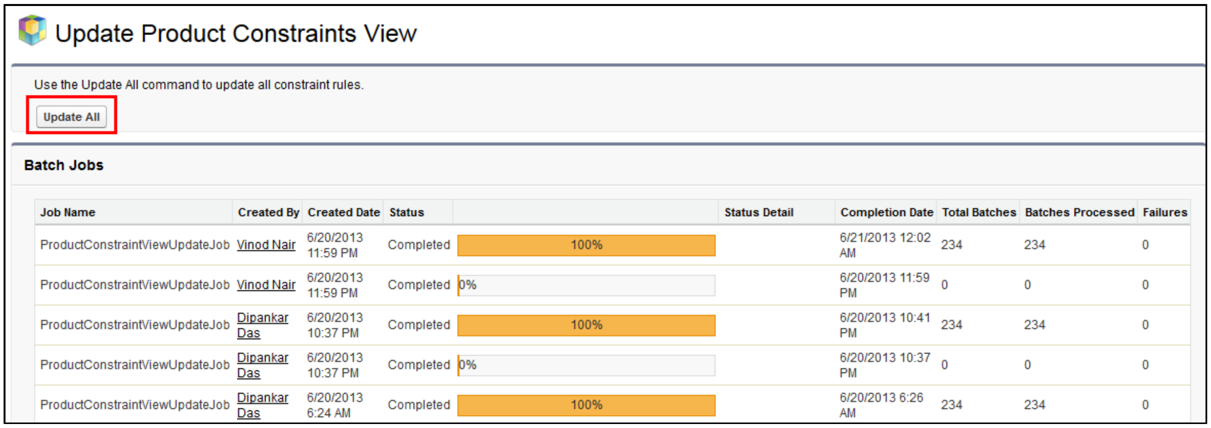
If the tab is still not available, as a workaround you can manually append apex /Apttus_Config2__ProductConstraint View to your URL in the address bar.



To run a constraint rule maintenance job

1. Go to the Constraint Rule Maintenance tab.

2. Click **Update All**.



The batch job is executed. The administration task is complete and an updated history for all the batch jobs is displayed. The key item to observe is Status. When Completed is displayed it means the job has run successfully, even if the percentage indicator remains at 0%.

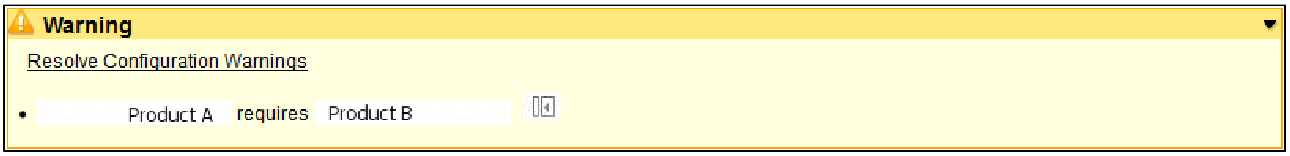
Setting up an Inclusion Rule

Inclusion rule use case: You have a product A and product B. You want to set up an inclusion rule such that when a user selects Product A, a warning message is displayed to include Product B.

To set up an inclusion rule

1. On the Constraint Rule Condition Detail page, in the Criteria section, from Product Scope, select *Product*.
2. In Product, enter *Product A* as the product and in the Match Conditions section, select Match in Primary Lines and click Save.
3. On the Constraint Rule Action Detail page, in Message, type *{0} requires {1}*. {0} denotes the condition product and {1} denotes the action product.
4. From Action Type, select *Inclusion* and from Action Intent, select *Show Message*.
5. From Action Disposition, select *Warning*.
6. From Product Scope, select *Product* and in Product, enter *Product B* as the product.
7. In the Match Conditions section, select Match in Primary Lines and click Save.

The inclusion type rule is saved and added. On the catalog page, when a user selects Product A, the system shows a warning message.



Auto-Inclusion with Line Item Updates

While creating auto-inclusion constraint rules, you can specify an Action Criteria based on which products are auto-included. For every new product auto-included, line items are added to the cart. You can now update the fields of the auto-included line items, such as, quantity, term, base price, and adjustment type. You can also update the fields based on condition defined in the Update Expression field.

> ### Note
>
> Line updates are only applicableonAuto-inclusion rules with the scope asproduct. Add only 10 line item expressions. Please consult with Apttus before adding more than 10 action expressions. There may be a more optimal way of approaching the requirement.

Scenario 1

If you purchase ten chargers or more, you can offer 2 USBs free. In this case, if the quantity of chargers in the cart is updated to >=10, then auto-include USB to the cart with the quantity set to 2, adjustment type set to % discount, and Adjustment value set to 100.

If the quantity of the chargers on the criteria line item is updated to less than 10 then auto remove Auto included USB line item from the cart.

Scenario 2

If you purchase a product with selling term greater than 2 years, you can auto-include 1 year of Premium Maintenance for free. In this case, if the Term of the product line item in the cart is updated to >=24, then auto include another line item in the cart for Premium Maintenance with term set to 12 months, start date set to start date of the criteria line item, adjustment type set to % discount, and Adjustment value set to 100.

If the term of the product on the criteria line item is updated to less than 24 then auto remove Auto included Premium maintenance line item from the cart for each Quantity.

Scenario 3

For every 10 chargers you buy, auto-include 2 chargers. In this case, if the quantity of chargers in the cart is updated to 56, then auto include another line item in the cart for chargers with quantity set to 10 (56/10 = 5), adjustment type set to % discount, and Adjustment value set to 100.

If the quantity of the chargers on the criteria line item is updated to 60 then the quantity of the Auto included charger line item in the cart is updated to 6 from 5.

If the quantity of the chargers on the criteria line item is updated to 45 then update quantity of Auto included charger line item in the cart to 4 from 5.

If the quantity of the chargers on the criteria line item is updated from 56 to less than 10 then auto remove Auto included charger line item from the cart.

To specify line item updates to Auto-Included Products

1. Navigate to **CPQ Console**, click **Inclusion Rule**. All the Inclusion Rules are listed.
2. Click **Edit Action Expression** next the **Message** field. The Edit Action Expressions popup appears.
3. Select the field you want to update and click the **Expression** field. The expression builder appears.
4. Specify the Expression using which the Line Item Field is populated. For example, you can populate the Adjusted Price as 100 percent for an Auto-included Line Item. You can also update the Line Item field based on rollup field you have defined. Specify the roll up field name.

Adding a product that triggers an auto-inclusion rule, adds another product as a line item whose fields can be populated using the Expression Builder.

Setting up an Exclusion Rule

Exclusion rule use case: You have a product A and product B. You want to set up an exclusion rule such that when a user selects Product A, a error message is displayed to exclude Product B.

To set up an exclusion rule

1. On the Constraint Rule Condition Detail page, in the Criteria section, from Product Scope, select *Product*.
2. In Product, enter *Product A* as the product and in the Match Conditions section, select Match in Primary Lines and click Save.

3. On the Constraint Rule Action Detail page, in Message, type *{1} is excluded on the addition of {0}*. {0} denotes the condition product and {1} denotes the action product.
4. From Action Type, select *Exclusion* and from Action Intent, select *Prompt*.
5. From Action Disposition, select *Error*.
6. From Product Scope, select *Product* and in Product, enter *Product B* as the product.
7. In the Match Conditions section, select Match in Primary Lines and click Save.

The exclusion type rule is saved and added. On the catalog page, when a user selects Product A, and then selects B. The system shows an error message that Product B is excluded in the addition of Product A.

Setting up a Validation Rule

Validation rule use case: You have a product A and you want to set up a validation rule such that when a user selects Product A and should not be able to enter more than 5 quantity.

To set up a validation rule

1. On the Constraint Rule Condition Detail page, in the Criteria section, from Product Scope, select *Product*.
2. In Product, enter *Product A* as the product and in the Match Conditions section, select Match in Primary Lines and click Save.
3. On the Constraint Rule Action Detail page, in Message, type *{0} cannot exceed 5 quantity*. {0} denotes the condition product.
4. From Action Type, select *Validation* and from Action Intent, select *Show Message*.
5. From Action Disposition, select *Warning*.
6. From Product Scope, select *Action Criteria*
7. In the Match Conditions section, select Match in Primary Lines and click Save.
8. On the Constraint Rule Detail page, scroll down to the Constraint Rule Actions related list and click Edit Criteria.
9. From Field, select *Line Item : Quantity*, from Operator, select *greater than*, and in Value, type *5*. Click OK.

The validation type rule is saved and added. On the catalog page, when a user selects Product A, and enters quantity as 6. The system shows an error message that Product A cannot exceed 5 quantity.

Quote/Proposal Management

Enabling Quote Collaboration in your org

Quote collaboration feature empowers your sales reps to work together on the same quote. A sales rep starts the configuration process, and assigns configuration request to other sales rep (called collaborator) where he needs help. The sales rep can assign multiple line items to a collaborator. A sales rep can also assign different line items to different collaborators.

The Quote Collaboration feature is useful for a large industrial equipment manufacturer, which has a complex quoting needs. The people who understand the products are located in different factories around the world. When a client requests a quote from a sales rep, the sales rep has to contact different factories and get configuration information. The problem is that the sales rep does not always know who is the right person for the job and does not have all the necessary support and collaboration. This burdens the quoting process by:

- Slowing it down.
- Making it more complex and expensive.
- Making it less accurate.

In such a scenario, quote collaboration enables the sales rep to collaborate while working on a single quote by assigning configuration tasks to different people and orchestrating that process. Then, the sales rep can merge the collaboration data to generate one final quote and send it to the customer.

To enable Quote Collaboration in your org

1. Add the **Collaboration** column by navigating to **All Tabs > Config Settings > Display Columns settings.** Choose the **Display Type** as *Cart Line Item* and **Flow** as *ngFlow*. Apart from the desired flow (e.g. ngFlow), the **Collaboration** column must be added to the settings for *Default* flow as well.
2. Specify the following collaborator fields API names in the **Setup > Develop > Custom Settings > Config System Properties > Manage > System Properties > View Cart Custom Fields.**
   - Apttus_Config2__CollaborationRequestId__c

- Apttus_Config2__CollaborationRequestId__r.Apttus_Config2__Status__c

3. Navigate to **CPQ Console > Maintenance Console > Query Field Metadata** and click **Refresh Field Metadata**.
4. Create Lookup Field Settings for User and Queue assignment.
5. Ensure that the Salesforce queue which needs to participate in the collaboration process supports **Product Configuration** and **Collaboration Request** objects. This is required to assign a collaboration request to the queue.

> - Ensure that you assign appropriate field level permissions for Collaboration Request object and Line Item object.
> - If the collaboration icon is not displayed on the Cart, ensure that **Collaboration** field's **Is Editable** property is set to *False* under Display Column Settings > Cart Line Item.

To enable Chatter on Collaboration Request

A chatter feed is enabled on each collaboration pop up. The feed is in the context of a single collaboration request and it can be accessed through a new tab on the collaboration pop up.

Ensure that you configure the following steps:

1. Go to **Setup > Build > Customize > Chatter > Feed Tracking**.
2. In the Object section, search for *Collaboration Request*.
3. In the Fields section, select **Enable Feed Tracking**.
4. Click **Save**.

Ensure that you have Quick actions such as Post, File, Link, and Poll on the Chatter Layout. You can verify these actions from **Setup > Build > Chatter > Publisher Layouts**.

To setup the Email Templates for Quote Collaboration

In order to send emails to the Sales rep and Collaborator for different actions involved in Quote Collaboration Lifecycle, you must setup the following email templates in your org.

Template Name	Description
Collaboration Task Assigned	When sales rep assigns the configuration to a collaborator, collaborator receives an email. The content of the email is controlled from Collaboration Task Assigned template.
Collaboration Task Completed	When collaborator submits his configuration, sales rep receives an email. The content of the email is controlled from Collaboration Task Completed template.
Collaboration Task Merged	When Sales rep accepts and merges the configuration submitted by the collaborator, collaborator receives an email. The content of the email is controlled from Collaboration Task Merged template.
Collaboration Task Updated	After assigning a configuration to the collaborator, when sales rep updates such configuration request, the collaborator receives an updated email. The content of the email is controlled from Collaboration Task Updated template.
Collaboration Request Cancelled	When sales rep updates the configuration request owner and re-submits the same configuration to someone else, the previous collaborator receives a cancellation notification email. The content of the email is controlled from Collaboration Request Cancelled template.

To setup the email templates, navigate to **Setup > Communication Templates > Email Templates** and create separate records for the above-mentioned templates.

Once you have created the above-mentioned templates, you need to create workflow rules for **Collaboration Request** object from **Setup > Build > Create > Workflow & Approvals > Workflow Rules**, which uses the appropriate template to send out the email notification.

Workflow Rule Name	Object	Evaluation Criteria	Rule Criteria
QuoteCollabNotifyOnCreate	Collaboration Request	Evaluate the rule when a	

Workflow Rule Name	Object	Evaluation Criteria	Rule Criteria
		record is created, and every time it's edited	OR (AND (ISCH (Apttus_Config2 (Apttus_Config2 AND (ISCHANG (Apttus_Config2
QuoteCollabNotifyOnUpdate	Collaboration Request	Evaluate the rule when a record is created, and every time it's edited	AND (NOT(ISC (ISBLANK (Apttus_Config2 NOT(ISCHANG (Apttus_Config2 ISPICKVAL(Apt 'Submitted'), ISC LastModifiedDa
QuoteCollabNotifyOnComplete	Collaboration Request	Evaluate the rule when a record is created, and every time it's edited	AND (ISCHANG (Apttus_Config2 (Apttus_Config2
QuoteCollabNotifyOnAccept	Collaboration Request	Evaluate the rule when a record is created, and every time it's edited	AND (ISCHANG (Apttus_Config2 (Apttus_Config2
QuoteCollabCancelNotificationOnChangeOfOwner			

Workflow Rule Name	Object	Evaluation Criteria	Rule Criteria
	Collaboration Request	Evaluate the rule when a record is created, and every time it's edited	AND (ISCHANGED(O ISCHANGED(L ISPICKVAL(PR Apttus_Config2_)
QuoteCollabNotifyOnUpdateAfterCompletion	Collaboration Request	Evaluate the rule when a record is created, and every time it's edited	AND (NOT(ISCHANG AND(ISCHANG (Apttus_Config2 ISPICKVAL(Apt "Submitted"), OR(ISPICKVAL (Apttus_Config2 ISPICKVAL(PR (Apttus_Config2)), ISCHANGED(L)

For the corresponding Workflow Rules, create the appropriate email alert action (email template, and the recipient) from the **Workflow Actions** section.

To enable Configure Products on the Collaboration Request Detail page

For the collaborator to start working on his configuration request, as an Admin you need to ensure that **Configure Products** button is available on the **Configuration Request Detail** page. You can use the following formula to create a custom formula filed for the Collaboration Request Detail page.

```
IF( ($User.Id == OwnerId),
HYPERLINK("/apex/Apttus_Config2__CollabRequestInline?id="&Id&"&flow=n
gFlow" , IMAGE("/resource/Apttus_Config2__Button_Configure", "Configu
re Products"),"_self"),
NULL
)
```

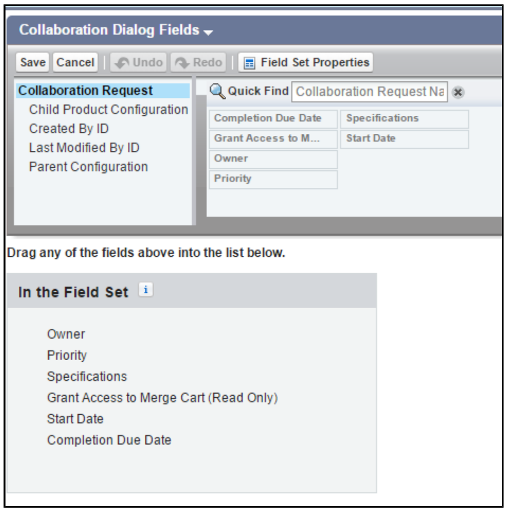
Configuring Collaboration Pop Up

You can specify the fields to be displayed on the Collaboration Pop Up dialog box. This configuration enhancement allows you to configure your own set of relevant fields on the assignment pop-up in order to capture adequate information before the request is sent or assigned to the collaborator to work on.

To configure the collaboration pop up Assignment Details:

1. Navigate to **Setup > Create > Objects > Collaboration Request**. This custom object persists the information for quote collaboration tasks and it is used to relate the parent configurations and its related child configurations.
2. Click **Edit** for **Collaboration Dialog Fields** Field Sets. You can control the fields that are displayed on the Quote Collaboration pop up **Assignment Details** tab.

3. Drag and drop the required fields in the **In the Field Set**.



4. Click **Save**.

You can configure the fields that you see on **Assigned Products** and **Completed Configuration** tabs from **Config Settings > Display Column Settings > Collaboration Parent Line Item** and **Config Settings > Display Column Settings > Collaboration Child Line Item** respectively. These two tabs are enabled after you have initiated the collaboration request with the collaborator.

Assigned Products tab shows the information of the collaboration request details as assigned by the sales rep. **Completed Configuration** tab shows the information of the collaboration request details as submitted by the collaborator. When you navigate to **Completed Configuration** tab, and click **More details** , you can see the submitted configuration in a read-only mode.

Configuring Default User or Queue Setup

By configuring default user or queue to be displayed on the collaboration popup dialog box, you can achieve a robust as well as accurate collaboration process, as the request assignment becomes faster and intuitive. The purpose of this feature is to enable enterprise organizations having a plethora of users and queues to filter and default their queues based on the product group, product family, account location and so on. In order for this feature to fulfill its goals and function properly, the right setup has to take place and link the desired user or queue to the filtering logic you want to use.

Ensure that following setup is configured successfully.

1. Set up a user or queue.

2. Set up Location on the quote: This quote location should carry over to the line items on the configuration. you can setup location or any other parameter. make sure that the line items have the location, if you want to filter based on the location.

> **Note**
>
> The field that you wish to base your filtering on, ensure that it is defined (whether it is a product family, product group, account location).

3. Set up Junction object - lookup junction
    - Include all fields that should be referred to. For example Queue ID (you can grab the ID from the URL), account location, product, product family, users, and so on.
    - You need to set this up for each user or queue separately.
    - After this setup, your user or queue is now tied to a location or a product. In order to expose this newly formed link, you need to configure the lookup filter.

> **Note**
>
> Default flag determines if this user or queue is default or not. Specify rest of the fields, you wish to link to your user or queue.

4. Set up **Lookup field settings**. This filter uses the information from the line item and matches it to the lookup junction object information. For example, if the line item's location is France, then all the queues that are defined with a junction object as located in France appears in the collaboration popup dialog box. You can default one of the French queues. In case if there are multiple queues are defaulted, the system only chooses the first one that gets defaulted. Other French queues which do not default are also displayed in the queue drop down and the sales rep can change the selection. Note that the default queue will be pre-selected.

Field	Description
Object Name	Specify the object name from which the lookup relationship is created.
Lookup Field Name	Specify the field name that shows the filtered results.
Display Columns	Specify the API names of fields to be displayed on the lookup dialog box, separated by a comma.
Filter Criteria	Specify the valid expression to further filter down the search results for a lookup.
Default Flag Field	Specify the API name of the specific default flag field on the junction object that stores the information whether the user or queue is default or not.
Source Field	Specify the API name of the field that contains the ID of the field that you are trying to lookup and display on the collaboration popup dialog box.
Source Object	Specify the junction object API name.

Setting up real-time cart notifications

Sales rep sees a notification popup on the top right side of the screen when collaboration request is submitted, completed, accepted and merged in the parent cart.

To enable this enhancement, after you upgrade to the November 2016 release, you must enter the following piece of code in the Developer Console/Workbench:


```
Apttus_Config2.NotificationFeedService.createPushTopic();
```

> You need to execute the above code only once and it will apply on all subsequent upgrades, unless you manually delete the PushTopic.

Enabling Cart Locking for Concurrent Access

Cart locking helps prevent data loss while a sales rep is working on a configuration. When a sales rep is working on a configuration (in edit mode), any other user who tries to access the cart at the same time will get redirected to a read-only view of the cart. The user in read-only mode can gain edit access only after the sales rep saves or finalizes or abandons or closes the cart.

A new custom setting, **Enable Cart Locking for Concurrent Access** has been introduced to handle the cart locking functionality. Available under **Setup > App Setup > Develop > Custom Settings > Config System Properties > Manage**, this custom setting enables your cart for concurrent access to other sales representatives.

A new field named **Current User** is added in the Product Configuration object which contains the ID of the user accessing the cart in the edit mode.

Enabling PDF Security on Generated Quote/Proposal Document

When you generate a quote in a pdf format, you can Enable PDF Security for the generated proposal document by following the configuration steps below:

To enable pdf security on the generated Quote/Proposal document

- You must set up *Proposal System Properties* for this feature. Go to **Setup > Develop > Custom Settings**. Click **Manage** next to *Proposal System Properties* list, then click **Edit** next to *System Properties* and do the following:
    1. Select the **Enable PDF Security** check box.
    2. Enter a desired value in the **PDF Owner Password**.

Refer Custom Settings > Proposal System Properties for detailed information.

1. Click and click **Admin.**
2. Do any one of the following:
   - Click **New** to create a new Admin entry.
   - Click **Edit** for an existing admin property *APTS_ProposalConfig*.
3. In the **Name** field, type *APTS_ProposalConfig* and in the **Value** field, type *XML*.
4. In the **Code** field, type the following code:

```
<ProposalConfig>
<ProposalSpec>
<CloneSpec>
<Excludes>
<ChildObject><Name>Apttus_Proposal__Proposal_Line_Item__c</Name>
</ChildObject>
</Excludes>
</CloneSpec>
</ProposalSpec>
<PDFSecurityDefault>
<CanPrint>true</CanPrint>
<CanCopy>true</CanCopy>
<CanChange>false</CanChange>
<CanAddNotes>true</CanAddNotes>
<CanFillFields>true</CanFillFields>
<CanAssemble>false</CanAssemble>
</PDFSecurityDefault>
</ProposalConfig>
```

5. All the tags defined over here are available in the *ProposalConfigSchema* static resource. To view this static resource, you may navigate to **Setup > Develop > Installed Packages > Apttus Proposal Management (Managed) > View Components > ProposalConfigSchema > View file**.
6. Click **Save.**

The proposal document generated in PDF format will now be secured with the system password.

Display Action Callback Class

A new custom callback class is supported to disable any of the action buttons on the cart. Following are few examples of custom buttons that you can disable.

- Generate
- Quick Save
- Approvals

```
global with sharing class APTS_QA_DisplayActionCallback implements
Apttus_Config2.CustomClass.IDisplayActionCallback{

List<Apttus_Config2.DisplayActionInfo> displayActions;
    global void start(Apttus_Config2.ProductConfiguration cart,
List<Apttus_Config2.DisplayActionInfo>displayActions){
        this.displayActions = displayActions;
}
    global void setActionProperties(Apttus_Config2.CustomClass.
ActionParams actionParam){
        for (apttus_config2.DisplayActionInfo action :
displayActions) {
            if(action.ActionSO.Apttus_Config2__ActionName__c == 'Aban
don'){
                action.isEnabled = false;
}
            if (action.ActionSO.Apttus_Config2__ActionName__c == 'Gen
erate'){
                action.isEnabled = false;
            }
        }
    }
}
```

The above callback class disables the **Abandon** and **Generate** buttons on the cart. You must specify this class by creating a new record under **Setup > App Setup > Develop > Apex Classes**.

Setting Up Quick Quote Mode

You can use Quick Quote mode to quickly select numerous products from the product selection page directly from an opportunity with just one click.

Quick Quote has enhanced search capabilities that include targeting standard and custom fields, advanced search, automatic pre-filters, saved list views, and can be sorted. When you have a long list of individual products, you can use the Select All function to select all individual products together.

> **Important**
>
> Only auto-inclusion constraint rules can be used with Quick Quote. Other constraint rules will be ignored.

To create quick quote custom button

1. Go to Setup > App Setup > Customize > Opportunities > Buttons, Links, and Actions and click New Button or Link.
2. Type a mandatory Label and a mandatory Name for the button. For example: Label = *Create Quick Quote* and Name = *CreateQuickQuote*.
3. Type a Description and from Display Type, select Detail Page Button.
4. Select a mandatory Behavior and a mandatory Content Source.
5. In the Code Snippet, type the following:

```
/apex/Apttus_QPConfig__OpptyQuoteCreate?id={!Opportunity.Id}
&recordTypeName=Proposal&priceListId={!Opportunity.
PriceListIdId__c}&method=quickAdd&useAdvancedApproval=true&defau
ltQuantity=1&defaultSellingTerm=1
```

6. Click Save.

The Quick Quote custom button is created.Add the Quick Quote custom button to the layout.**To add the quick quote button**

You must have already created a Quick Quote custom button.

1. Click Page Layouts and click Edit for the page layout to which you want to add the Create Quick Quote button.
2. From **Buttons** in the Layout configuration window, select **Create Quick Quote** and drag and drop it onto the page layout.
3. Click **Save** in the Layout configuration window.

Create Quick Quote button is added and saved onto the page layout.Click the Create Quick Quote button to select numerous products and proceed to create a quote.

Customizing The Visualforce Pages

You can configure which standard columns and actions are available in the catalog, attribute, bundle, and shopping cart pages, the style they use, and the order in which they are displayed. Additionally, you can create your own custom fields and actions and display them in these Visualforce pages.

The Visualforce pages are used for product selection, select attributes, options, and in the cart page change quantities, apply discounts, and more for your quote/proposal. If there are default fields and actions that are not required, or new fields you want to add, you can make those changes via the CPQ *Config Settings* tab.

- Customizing Display Columns
  The Display Column Settings enable you to display only those key fields needed to quickly and easily configure your quote/proposal.
- Customizing Display Actions
  The *Display Action Settings* page provides a simple interface for selecting actions, applying styles, and reordering them. You can even create and use your own custom action.
- Auto-synch Cart Lines
  The Custom Actions feature has been enhanced to auto-synch cart lines with proposal lines and navigate to other pages.

Customizing Display Columns

The Display Column Settings enable you to display only those key fields needed to quickly and easily configure your quote/proposal.

The *Display Column Settings* page provides a simple interface for selecting fields, making fields editable, applying styles, and reordering them.

To display different number type values as expected, you can add one of the following styles to cart line item fields:

aptCurrency	This will prepend the appropriate currency symbol for the org's locale and retain two digits after the decimal place for the field value. You can change the number of digits using the Currency Field Precision in Config System Properties.
aptPercentage	This will retain two digits after the decimal place (if present) and append the percentage symbol to the field value. You can change the number of digits using the Percentage Field Precision in Config System Properties.
aptQuantity	This will retain two digits after the decimal place (if present). You can change the number of digits using the Quantity Field Precision in Config System Properties.

The following style classes have to be applied in pair to the dynamically selected adjustment column and its value column. i.e., Adjustment Type and Adjustment Amount columns.

- aptAdjustmentType
- aptAdjustment

The formatting for the **Adjustment Amount** field will change dynamically based on the **Adjustment Type**. This is achieved by applying available style classes to the adjustment based on the value in the pick list of the adjustment type.
Note: For changes to the shopping cart beyond those described here, you would need to create custom pages of the Shopping Cart (*CartView* and *CartFinalize* Visualforce pages) and write your own Apex code to further control what fields are displayed and how the page is organized.

> In order to use PULL promotions on the Cart, you must follow the procedure mentioned below with **Display Type** = *Apply Promotions*.

To customize a display column

1. Click (All tabs) and click **Config Settings**.
2. Click **Display Column Settings**.

3. To create or modify columns on one or more of the following, select the view from the **Display Type** picklist.
    - Cart Line Item
    - Cost Line Item
    - Cart Total Item
    - Installed Product
    - Agreement Price Rule
    - Agreement Price Tier
    - Adjustment Line Item
    - Usage Price Tiers
    - Apply Promotions
    - Option Line Items
    - Price Ramp
    - Tiered Price
    - Related Price
    - Mass Update
    - Collaboration Parent Line Item
    - Collaboration Child Line Item
    - Line Item Key Metrics
    - Line Item Waterfall
    - Summary Group Key Metrics
    - Summary Group Waterfall
4. Select your desired flow from the **Flow** picklist.
    - Default
    - ngCPQ
    - ngFlow
5. The **Sequence** column is system-generated and based on your **Display Type** selection in Step 3, the **Display Type** column displays the line item name.

6. In the **Field Name** column, select a field you want to display as a column.Note: You should not add *Option* as a column as Product and Option columns are identical in the CartDetailView page.
7. To make the field name column editable, select **Is Editable**. Clearing the **Is Editable** check box, makes the field read-only.Note: If you select **Display Type** as *Installed Product*, the Is Sortable check box is displayed. To make the field name column sortable, select the Is Sortable check box.Note: For **Selling Term**, and **Selling Frequency** columns to be editable, apart from selecting the **Is Editable** check box, ensure that **PriceType** for the corresponding product is set as either *Recurring*, *Usage*, or *Included Usage*. Additionally, for **Selling Frequency**, ensure that **Allow Proration** check box is selected on the price list item.
8. In the **Style** column, type values separated with a colon to change the default appearance of a field. Example 1: If you want to resize all the columns in the cart, type *width: 30px;text-align:left*. Example 2: If a product name is lengthy and is not displayed properly on the shopping cart page, type *text-align: left; width: 300px; max-width: 300px; display: block;white-space:0px;*. The product name is wrapped and displayed completely on the cart page.
9. In the **Style Class** column, type one of the following to change the way dynamic columns are displayed:
   - *aptCurrency*
   - *aptQuantity*
   - *aptPercentage*
   - *aptAdjustmentType*
   - *aptAdjustment*

   Note: These style classes apply two decimal points to the fields accordingly. You can change the number of decimal places from **Config Settings > System Properties** using the *Currency Field Precision*, *Percentage Field Precision*, and *Quantity Field Precision*.
10. In the **Header Style** column, type values separated with a colon to change the default appearance of the header of a column. For Example: *width:200px;*
11. You can also do one or more of the following:
    - To add or delete a row, use the and icons.
    - To re-order the rows, use the and icons.
12. Click **Save.**

The columns are displayed as configured in the Display Columns Settings.

Customizing Display Actions

The *Display Action Settings* page provides a simple interface for selecting actions, applying styles, and reordering them. You can even create and use your own custom action.

In order to create custom actions, you must first create a custom label.

To create a custom label

1. Go to **Setup > Create > Custom Labels** and click **New Custom Label**.
2. Type a mandatory **Short Description** and type a mandatory **Name.**
3. In **Categories**, type comma-separated values that can be used in filter criteria when creating custom label list views. For example: Pages, Label, Components, and more.
4. In **Value**, type mandatory text in the form of a brief description or copy and paste the field name from Step 1.
5. Click **Save.**

A custom label is created and saved.

To customize display actions

If you want to create and use custom actions, you must have an existing custom label for the action.

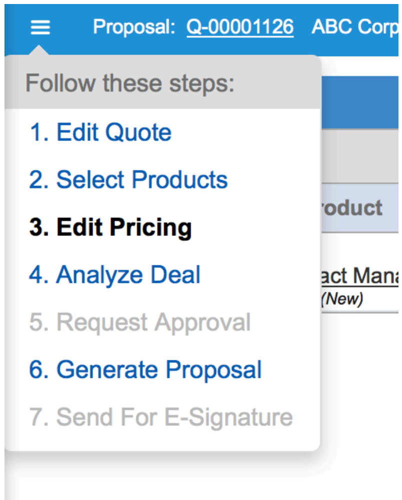
1. Click (All tabs) and click **Config Settings**.
2. Click **Display Action Settings**.
3. From **Display Type**, select the visual force page where you want to place the custom action.

   > For the angularJS UI, select the Attribute Page from **Display Type** to configure attributes as well as options. Bundle Page display action settings are deprecated and they are replaced with Attribute Page display action settings.

4. The **Sequence** column is system-generated and based on your Display Type selection in Step 3, the Display Type column displays the line item name. The Action and Action Label Name display the default actions.

5. You can create custom Action button, where the **Action** column name is CustomAction <number> and in **Action Label Name** column, type the field name of the custom label you created. For example: Action = CustomAction1 and Action Label Name = Edit_Quote.
6. From the **Display As** column, select one of the following:
   - To create and set up the action as an Action button, select *Action*.
   - To create and set up the action as a Task menu item in the Status bar, select *Task*.
   - To create and set up the action as an Action button and also as a Task menu item, select *Action and Task*.
   - To create and set up the action as a hyperlink in the Navigation bar, select *Nav Link*.
   - To create and set up the action as a Help menu icon, select *Help*.
7. From the **Action Area** column, select one of the following to position the action:
   - *Center* - For center alignment
   - *Left* - For left alignment
   - *Right* - For right alignment
   - *More* - To add as an item under the More action button
8. In the **Action Style Class** column, type one of the following available style class name for the action.

Style	**Description**
apt-selected-btn	

Style	Description
	Type this to display the button in the drop-down menu with a white background and black font color. This also indicates the step that an end user is on. 
apt-left-btn	Type this to display an arrow pointing left on the button.
apt-right-btn	Type this to display an arrow pointing right on the button.

Note

If you want to use your own style, create a CSS file including all of the custom styles you want in the shopping cart and upload it as a static resource. Type the name of the CSS file in **Config Settings > System Properties > CSS Override**. This enables you to use the styles defined in the CSS file as opposed to out-of-the-box styles.

9. In the **Action Page** column, in **Action Page**, type the custom Visualforce page name and in **Action Params**, type the parameter that you want to pass for your custom page. For example: {!Apttus_QPProposal__ProposalID__c.ID} If you leave the **Action Params** field

blank, by default, the system includes the ID, Business Object ID, and Config Request ID as the dynamic parameter. Example URL: .....CustomVFPage?id=a1Oe00000018DLUEA2&retId =a0ee0000003GUCfAAO&businessObjectId =a0ee0000003GUCfAAO&configRequestId=a1ie0000000kbjTAAQ

10. From the **Behavior** column, to specify the action page behavior, select one of the following:
    - To open within the same window, select *Self*.
    - To open in a new window, select *New Window*.
    - To open in a dialog, select *Dialog Window*.
11. From the **Action Type** column, select one of the following:
    - To redirect to an another page using the custom button and not show the spinner, select *Quick Redirect*. This option ignores the unsaved changes made by the customer before clicking the custom button.
    - The other options, such as *Save, Quick Save*, *Submit for Approval, and Generate* perform the same function as the out-of-the-box action buttons.
12. To enable or disable actions, select or clear the **Is Enabled** check box.
13. To enable permanent display of the action irrespective of the setting in Step 12, select Always Display check box.

Scenario	Display Action
Is Enabled = True and Always Display = True	Action is displayed.
Is Enabled = False and Always Display = True	Action is displayed but inactive. You cannot click the action.

14. Do one of the following:
    Click **Save** to save the changes you made in this page.
    Click **Load Default Classic Settings** to load system default Classic Settings.
    Click **Load Default Angular Settings** to load system default Angular Settings.

The action is added to the list of actions to be displayed on the Visualforce pages. At any given time, if you want to disable the custom action, go to the Display Action Settings page and clear the **Is Enabled** check box.

Auto-synch Cart Lines

The Custom Actions feature has been enhanced to auto-synch cart lines with proposal lines and navigate to other pages. When Auto-synch is turned on, the system does validation for errors and auto-synchs the cart lines with the Proposal Lines. You can view the synched line items in the Quote/Proposal detail page under the Line Items related list. Auto-sync is done when following conditions are satisfied:

- No error produced by validation callback
- No Constraint Rules error
- No line items in price pending state
- No Must Configure product is in configuration pending state

To configure custom actions for auto-synch

1. Click (All Tabs) and click Config Settings.
2. Click Display Action Settings, and scroll down to add a custom action button.
3. Type an Action Label Name, and from the Display As list, select *Action*.
4. Select an Action Area, and in Action Style Class, specify a style class.
5. In Action Page, type the name of the Visualforce page that appears when the action button is clicked.
6. In Action Params, type *autoSync=true*.
7. In Behavior, select an option of how you want the action page to open.Note: The auto-synch custom actions should not have action type (Save, Quick Save, Generate) specification.
8. To enable the custom Action button, select Is Enabled.
9. To always display the custom Action button whether enabled or disabled, select Always Display.

A custom action is created with auto-synch capability.

Multiple Layouts For Page Columns And Actions

As an administrator, you can now vary the Catalog, Options, Attributes, Installed Products and Cart pages that an end user interacts with as well as the data and actions within those pages based on the end user profile and other business rules. This is made possible by introducing the concept of Flows. Flows are groups of pages that are assigned to each step in the CPQ process. Earlier the only option to set up the Visualforce pages was through custom settings.

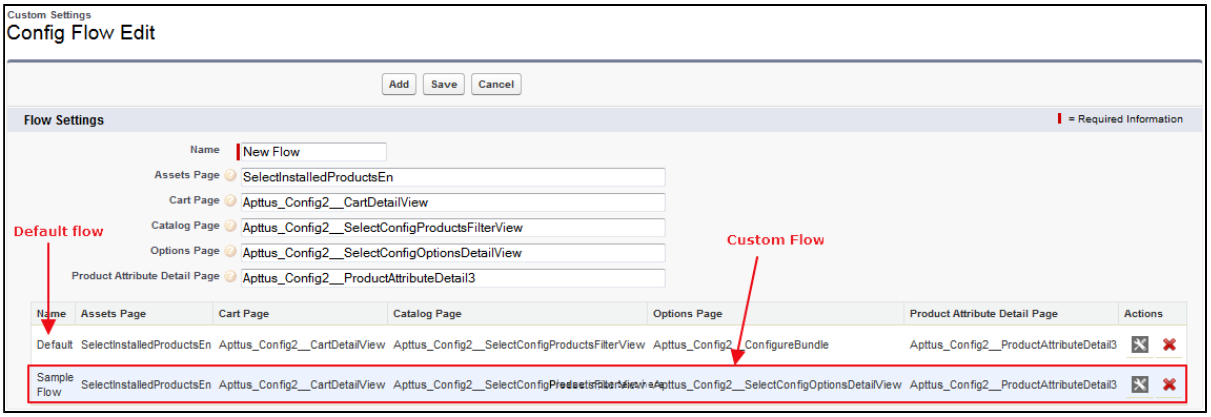
You can now use Flow Settings to assign custom or out-of-the-box pages to each step and configure the fields and actions that are visible within each page. You can then create criteria that would determine the Flow which in turn determines the pages that are displayed when an end user clicks the Configure Products button. These flow settings once setup, override the Visualforce pages setup in custom settings.

- To create flows using flow settings
- To set up the configure products button to use the flow
- To set up columns and actions for flows

To create flows using flow settings

1. Click ✚ and click **Config Settings**.
2. Click **Flow Settings**.
3. In **Name**, type a name for the flow. For example: Sample Flow
4. Type the Visualforce pages in the following fields:
    - Assets Page
    - Cart Page
    - Catalog Page
    - Options Page
    - Product Attribute Detail Page

   Hover the mouse over ? to see valid values for each field.
5. To add the new flow, click **Add**. An entry is added to the list below.



Default - When there is no flow created, the system uses the Default flow.

Custom Flow - In this example, Sample Flow is the custom flow created.

6. Click **Save**.

A new flow is added and saved. Use the flow name you just created in the parameter to pass on for the Configure Products button.

To set up the configure products button to use the flow

You must have an existing flow.

1. Go to **Setup > App Setup > Create > Objects**.
2. Scroll down to select the Quote/Proposal object and from the **Custom Fields & Relationships** related list, select **Configure Products**.
3. Click **Edit** and insert the flow name in the parameter as below.

```
IF ( LEN(Apttus_QPConfig__PriceListId__c ) > 0 , HYPERLINK("
/apex/Apttus_QPConfig__ProposalConfiguration?id=" &Id+'&flow=Sam
ple Flow', IMAGE("/resource/Apttus_QPConfig__Button_Configure",
"Configure Products"),"_self"), NULL)where flow=<name of the
flow you created>
```

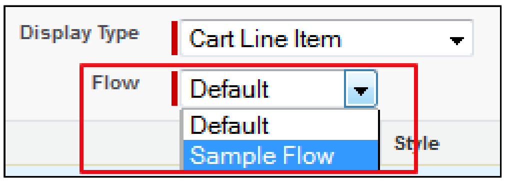
4. Click **Save**.

The parameter is saved for the Configure Products button. On the Quote/Proposal page when you click the Configure Products button, all the pages appear as defined in the flow. You can modify the columns and actions that you want to use with the flow you created.

To set up columns and actions for flows

You must have an existing flow.

1. Click ✚ and click **Config Settings**.
2. For setting up columns or actions or both for the custom flow, do one or more of the following:.
    - For columns, click **Display Column Settings**
    - For Actions, click **Display Action Settings**
3. Select a **Display Type** and from **Flow**, select the flow you created.



4. Click **Save**.

The custom display columns and actions are set for the flow you created. All the Visualforce pages that you set up in the flow are displayed with the columns and actions set up in the config settings. If you select Default flow, the system displays the columns and actions set up in the Default flow. Clicking the Load Default Settings button will override all the columns and action settings to out-of-the-box settings in the Default flow.

Creating Promotional Banners

When a user transitions between pages (such as from the Catalog page to the Shopping cart), a progress bar is displayed. The text and image can further be customized to provide relevant information to the user such as promotional or informational messages.

You can configure the following banners:

- Progress Bar: Enables you to display text underneath the Progress Bar during pricing. You can specify multiple text messages to be shown and the sequence in which they appear by specifying multiple progress bar settings.

- Marketing Banner: Enables you to display image or text underneath the banner text (if specified) or progress bar. You can specify multiple marketing banners to be shown by specifying multiple Marketing banner settings. The sequence in which Marketing Banners appear is completely random.

To create banner settings

If you want your Banner message to be of type text ensure that you create a custom label for the same. If you want to specify an image in the banner, ensure that an image is saved as a static resource in your org.

1. Go to Custom Settings > Banner Settings > Manage > New.
2. Specify a name for the setting.
3. Do any one of the following:
    - To specify a banner of type text, enter the BannerMessageContext as Progressbar.
    - To specify a banner of type image, enter the BannerMessageContext as Marketing.
4. Specify the BannerPageContext as Pricing. The banner appears when the pricing callback is invoked on click of a button and the user is navigated to a new page.
5. Do any one of the following:
    - If the BannerMessageContext is Progressbar, specify the custom label name as the BannerSource.
    - If the BannerMessageContext is Marketing, specify the URL of the image stored in static resources as the BannerSource.
6. Do any one of the following:
    - If the BannerMessageContext is Progressbar, enter CustomLabel as the BannerType.
    - If the BannerMessageContext is Marketing, enter ImageURL as the BannerType.
7. Specify the sequence in which the banner appears. The sequence you specify for the textual banner, is the sequence in which the banner messages appear to the user. The sequence field is applicable only when the BannerMessageContext is Progressbar.
8. Click Save.
9. Repeat Steps 1-8 to create multiple textual or image banners to be shown to the user. The Progressbar banners appear in the sequence in which they are defined. The Marketing banners appear randomly.

The banner settings are saved and created successfully.

Hiding the Clone Icon for a Cart Line Item

You can now choose to hide the clone icon for a specific cart line item. If you select the **Hide Copy Action** check box at **Config Settings > System Properties**, it hides the clone icon for all the line items in the cart. For a Line Item Object there is a new field - **Hide Copy Action** of type *Boolean* which enables you to hide the clone icon for a particular line item.

Scenario: Suppose you have configured two products A and B and you want to hide the clone icon whenever the user selects a product with an attribute of color *Red*. Create a workflow rule with a criteria Line Item: **Color** equals *Red* and the workflow action as Field Update on the Line Item Object and the Field to Update as Line Item: Hide Copy Action and set the new field value as *True*.

Following are the values that you can set in the **Hide Copy Action** check box with the subsequent results:

Hide Copy Action System Property	Workflow Rule Field Update Value	Result at the Line Item
True	True	Clone icon hidden
True	False	Clone icon hidden
False	True	Clone icon hidden
False	False	Clone icon displayed.

To hide the clone icon for a cart line item

1. To hide all the line items, navigate to **Config Settings > System Properties**, and select the Hide Copy Action check box.
2. Create a workflow rule for a Line Item Object with an appropriate rule criteria.
3. Set the Workflow action as **Field Update** and set the field to update on **Line Item Object** and the **Field to Update** as *Line Item*: *Hide Copy Action* and set the new field value to *True*.

The clone icon is hidden for a cart line item based on the **Hide Copy Action** value in System Properties and the Workflow Rule Action.

Retaining the order of Line Items in a Generated Quote /Proposal Document

The sequence of the Line Items on the Cart depends on the addition and configuration of the products on the Catalog and Configuration pages. Using the drag and drop feature, a Sales Rep can rearrange the Line Items on the Cart and retain the rearranged sequence in a generated Quote /Proposal document.

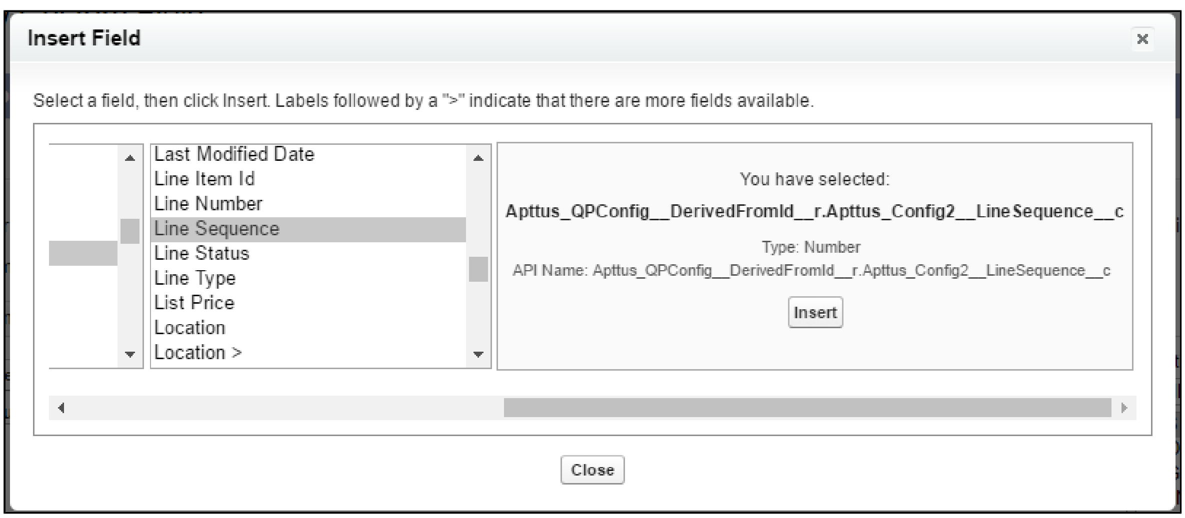
To retain the order of the Line Items from the Cart to the Generated Quote/Proposal document, you must perform the following procedures:

- Create new formula fields for Line and Option Sequence
- Create an Admin entry for Line and Option Sequence

Creating new formula fields for Line and Option Sequence

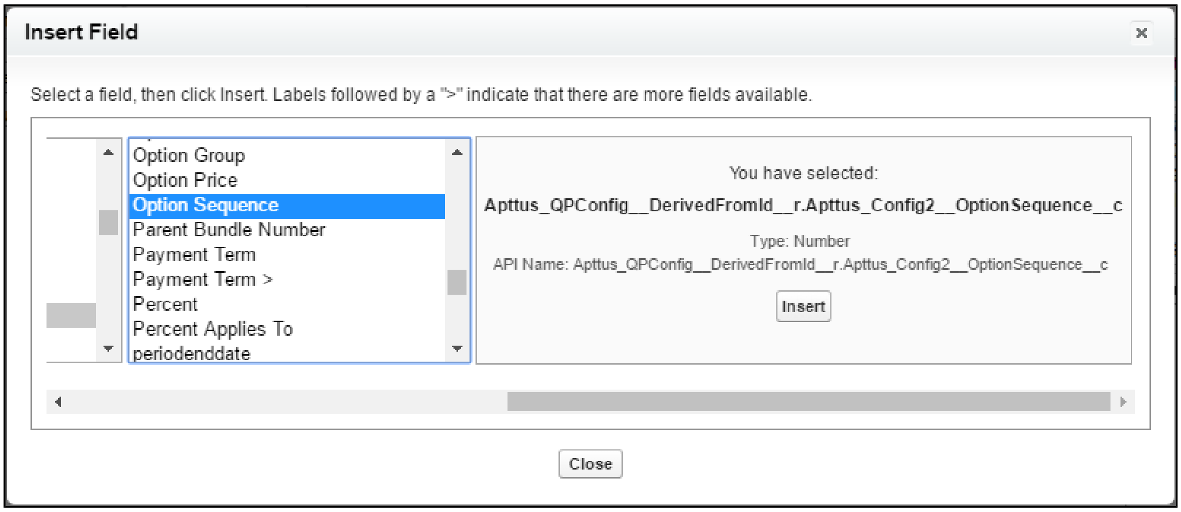
You must create two new custom formula fields for Proposal Line Item object, which must be derived from **Line Sequence** and **Option Sequence** fields of Line Item object.

1. Go to **Setup > Create > Objects** and click **Proposal Line Item**.
2. In the Custom Fields & Relationships related list, click **New** and create a new Formula field.
3. Enter the field Label as *Sequence Line* and choose the **Formula Return Type** as *Number*.
4. Choose the Advanced Formula tab and build the formula field using the flow- **Proposal Line Item > Derived From > Line Sequence**.



5. Grant the necessary Field Level Security, assign the new custom field to the Page Layouts and click **Save**.

6. Repeat step: 2 to 5 for creating a new custom formula field for *Sequence Option* field, which will be derived from the formula- **Proposal Line Item > Derived From > Option Sequence**.



Creating an Admin entry for Line and Option Sequence

1. Click and click **Admin.**
2. Do any one of the following:
   - Click **New** to create a new Admin entry.
   - Click **Edit** for an existing admin property *APTS_ProposalConfig*.
3. In the **Name** field, type *APTS_ProposalConfig* and in the **Value** field, type *XML*.
4. In the **Code** field, type the following code:

```
<ComplyConfig>
<SortSpec>
<SortObjects>
<SortObject>
<Name>Apttus_Proposal__Proposal_Line_Item__c</Name>
<SortFields>
<SortField>
<Name>SequenceLine__c</Name>
</SortField>
<SortField>
<Name>SequenceOption__c</Name>
```



```
</SortField>
</SortFields>
</SortObject>
</SortObjects>
</SortSpec>
</ComplyConfig>
```

Depending upon your business object (Quote/Proposal or Agreement), you can enable this for ProposlConfig or ComplyConfig respectively.

5. All the tags defined over here are available in the *ProposalConfigSchema* static resource. To view this static resource, you may navigate to **Setup > Develop > Installed Packages > Apttus Proposal Management (Managed) > View Components > ProposalConfigSchema > View file**.
6. Click **Save.**

Now when you generate a Quote/Proposal document, the rearranged sequence of the Line Items is reflected.

Setting Up Guided Selling

Guided Selling is a process that helps a sales representative to choose the product best fulfilling a customer's needs. The goal of guided selling is to provide an inexperienced sales representative the way to figure out which product to sell to a customer.

During the quoting process, the sales representative answers a set of questions around the customer’s needs and based on their answers, the system recommends products. This way a sales representative can quickly select the right product to meet a customer’s needs.

To configure and set up Guided Selling, you must do the following:

- Prepare a list of questions and answers to be displayed on the guided selling page. You should decide which questions to ask the customer and based on which product characteristics you want to filter the products.
- Create a custom field in Search Attribute Value object for each question.
- In Product object, create custom fields that you created in the Search Attribute Value object. Note: API Name must be the same as Search Attribute Value object.
- Create a flow using Salesforce Flow Designer.
- Create a guided selling visualforce page.
- Define Guided Selling Rules.

Creating Custom Fields in Search Attribute Value Object

You must create custom fields in the search attribute value object to correspond to questions.

Search Attribute Value is the object that stores the answers to the questions that an end user gives everytime they go through guided selling. You need to create a custom field in *Search Attribute Value* object for each question. The value stored in the question field is the actual answer given to the question by the representative.

Example: You have a question *Does customer need 24X7 toll-free support?*. In Search Attribute Value object, create a field *Does customer need 24X7 support*.

Note

Name does not have to match the question.

The answer the sales representative gives to the question is stored in the custom field.

Creating a Sample Flow Using Flow Designer

Salesforce Flow Designer helps you create a flow by defining questions and answers that you may want to allow the end user for filtering products.

Below is an example on how to setup a simple flow using Salesforce Flow Designer. For detailed information on various ways to create a flow, see Creating a Flow.

Say you want the end user to answer a question and based on the selection, you want the system to recommend products. Here is an example:

- Add a variable
- Add a Screen Element
- Add a Record Update Element
- Set a Start Element

To create a flow

Navigate to **Setup > Create > Workflow & Approvals > Flows**.

1. Click **New Flow**.
2. From the Palette in the left pane, click and drag a Screen element onto the canvas to configure it.
3. From the Palette in the left pane, click and drag a Record Update element onto the canvas to configure it.
4. Click **Save.**
5. In **Flow Properties**, enter a mandatory **Name** and press TAB. The Unique Name is auto-populated.
6. Enter a **Description** and click **OK.**

A flow is created and saved.

After the flow is created, to make the flow active, click **Activate.**

To add a variable

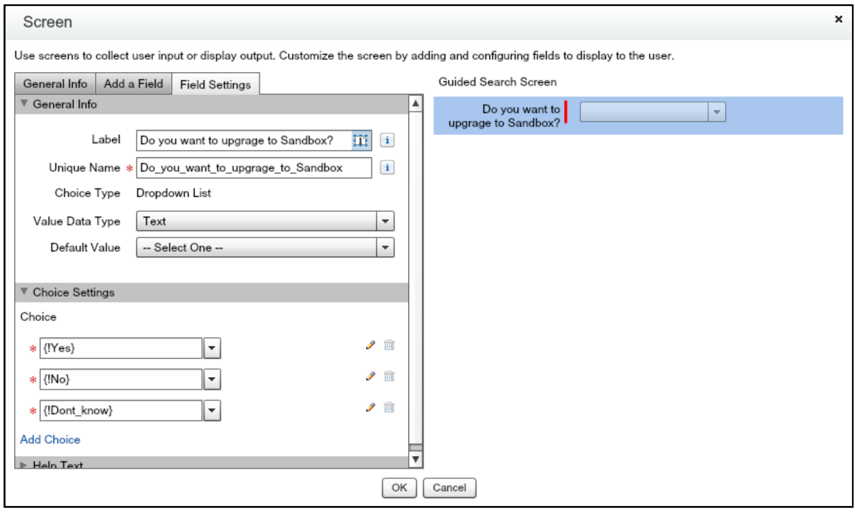
Navigate to **Setup > Create > Workflow & Approvals > Flows**.

1. Go to the **Resources** tab and from **CREATE NEW**, double-click *Variable.*
2. In **Unique Name**, type *InterviewId*.
3. Enter a **Description.**
4. From **Input/Output Type**, select *Input and Output*.
5. Click **OK.**

InterviewId is added as a variable.

To add a screen element

1. Enter a mandatory **Name** and press **TAB**. The Unique Name field is auto-populated.
2. Click the **Add a Field** tab and double-click Dropdown List. A Dropdown List field appears in the preview pane.
3. Double-click the Dropdown List field in the preview pane to configure it.
4. On the **Field Settings** tab, enter a mandatory **Label** and press **TAB**. The Unique Name field is auto-populated. For example: Label is *Do you want to upgrade to Sandbox?*, the Unique Name will be *Do_you_want_to_upgrage_to_Sandbox*.
5. In the **Choice Settings** section, next to **Select Resource** click .
6. Expand the **CREATE NEW** section, and select **Choice.**
7. Enter a mandatory **Label** and press **TAB**. The Unique Name is auto-populated.
8. In **Stored Value**, type *Yes* as the answer choice and click **OK**. To enter more than one choices, repeat steps 7 through 10. For Example: Enter *No* and *Don't know* as another answer choices.
9. Verify that your Screen overlay looks like this, and click **OK**.



The screen element is added.

To add a record update element

You must have created a screen element.

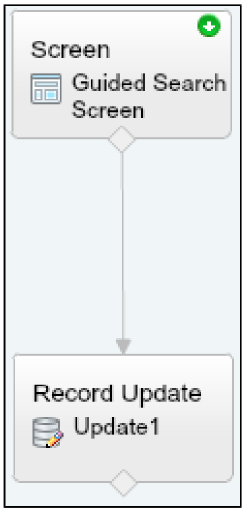
1. From the Palette in the left pane, click and drag a Record Update element onto the canvas.
2. Enter a mandatory **Name** and press **TAB**. The Unique Name field is auto-populated.
3. From **Update > Custom**, select *Apttus_Config2__SearchAttributeValue__c*.
4. From **Field > Standard**, select *Id.*
5. From **Operator**, select *equals*.
6. From **Value > Variables**, select *InterviewId.*
7. Under Update record fields with variable, constant, input, or other values, from **Field > Custom**, select the Custom field you created.
8. From **Value > Screen Choice Fields**, select the screen choice you created.
9. Click **OK.**

A record update is added.

To connect elements and set a start element

You must haved added the element to the flow.

1. Click and drag an arrow connector from the Screen element to the Record Update element.



2. To set the screen element as the start element, click .

The elements are connected and a flow chart is ready with a start element.

A Visualforce page is required to be created to display the flow you just created using the Flow designer. Create a Guided Selling Visualforce page to display the flow.

Creating a Guided Selling Visualforce page

A Visualforce page needs to be created to display the flow you just created using Flow Designer.

To create a guided selling Visualforce page

Go to **Setup > App Setup > Develop > Pages**.

1. Click **New.**
2. Enter a mandatory **Label** and a mandatory **Name.**

   There should be no spaces in **Name**. Example: *NameWithNoSpace*.
3. In your code for the visualforce page, enter the **Interview Name**.

   The interview name is the Flow Name.


```
<flow:interview name="Guided_Selling_Flow_Test" buttonLocation="
bottom" finishLocation="{!$CurrentPage.URL}&interviewId={!
InterviewId}&step=Option"  reRender="dummy" rendered="{!
IsBundleStep}" >
<apex:param name="InterviewId" value="{!InterviewId}" />
</flow:interview>
```

4. Click **Save.**

You can now define guided selling rule.

Defining the Guided Selling Rule

Guided Selling Rules map the search attributes in Apttus to product attributes for filtering.

To define the guided selling rule, you must create a guided search rule with a guided search rule entry and a guided search rule filter. The system filters the products based on the answers to the questions.

For example: In the **Search Attribute Value** object, you have a created a custom field **Number of users for the server**. You want the system to filter the products on their **Number of users** field.

- To define a guided selling rule
- To add a guided search rule entry
- To add a guided search rule filter

To define a guided selling rule

You must have already created custom fields in the **Search Attribute Value** object.

Navigate to the CPQ Console tab.

1. From the Rule Management section, click Add Guided Selling Rule.
2. In the New Guided Selling Rule section, type a mandatory Rule Name.
3. Type a Description and enter a mandatory Category.
4. In **Guide Page**, enter the guided selling Visualforce page name.
5. To make the guided selling rule active, select **Active**.
6. In **Effective Date**, enter a date from when the rule is effective.

7. In **Expiration Date**, enter an expiry date for the rule.
8. Click **Save**.

A guided selling rule is created and saved.You can now add guided search rule entries.**To add a guided search rule entry**

You must have already defined a guided selling rule.

1. In the Guided Search Rule Entry section, select a Question.
2. Select an Operator and type a Value.
3. To add more than one entry, click **Add Entry**.
4. Click Save.

Guided search rule entry is added.You may want to add guided search rule filters.**To add a guided search rule filter**

You must have a guided selling rule defined along with a guided search rule entry.

1. In the Guided Search Rule Filters section, select a Field and select an Operator.
2. From Map To, select **Question** or **Value**.
3. If you select Question in Step 2, you can see the list of fields in the Search Attribute Value object and select one of these fields. -OR- If you select Value in Step 2, you can enter a value in the Question / Value column.
4. To add more than one filters, click **Add Filter**.
5. Click **Save**.

Incentive Management

Apttus CPQ now comes with an integrated Promotions package that you can use to incentivise the sales of products in your inventory. Incentives aim to provide value for money f or your customers and contribute to better sales for your organization.

If you are a marketing manager or are in a role that is responsible for promoting some products over others, you can use incentives to,

- increase sales
- increase revenue margins
- enhance customer retention
- clear your stock

To achieve any one or all of these objectives, you can use two types of incentives, Promotions and Rebates.

A promotion is a marketing technique that you apply to reduce the list price of a product or a service. You can create such a promotion and restrict the scope, limit, and benefits so your sales representatives apply this promotion to specific products, for specific customers, and for a limited period.

A Rebate on the other hand is an incentive that your customers receive after buying the promoted product. Mail in rebates is the most common type of rebate where you send out a check to your customers for the decided amount. In this case, there is no reduction in the list price of the product or service that you promote.

The promotions package is available as an add-on with the CPQ license and package. To install Apttus Promotion Management, see Installing Apttus CPQ.

See the Promotions Workflow to understand how Promotions work.

Promotions Management

Apttus CPQ now comes with an integrated Promotions package that you can use to incentivise the sales of products in your inventory. Incentives aim to provide value for money f or your customers and contribute to better sales for your organization.

If you are a marketing manager or are in a role that is responsible for promoting some products over others, you can use incentives to,

- increase sales
- increase revenue margins
- enhance customer retention
- clear your stock

A promotion is a marketing technique that you apply to reduce the list price of a product or a service. You can create such a promotion and restrict the scope, limit, and benefits so your sales representatives apply this promotion to specific products, for specific customers, and for a limited period.

The promotions package is available as an add-on with the CPQ license and package. To install Apttus Promotion Management, see Installing Apttus CPQ.

See the Promotions Workflow to understand how Promotions work.

Using Promotions

In the Promotion Management section of the Apttus CPQ Administrator guide, you can quickly find out how Promotions Management works and how you can manage your organization's business requirements with respect to Promotions.

If you are a Marketing Manager, you will learn to

- Create and Edit Promotion plans
- Submit Promotions for Approval
- Approve or Reject requests for approval
- Analyse the impact and effectiveness of Promotions on Quotes and Orders

If you are a Marketing Executive, you will learn to

- Analyze Promotion Impact of Quotes / Orders
- Analyze Promotion Effectiveness

If you are a sales representative, see the Applying a Promotion section of the Apttus CPQ User guide to learn how to

- View Available Pull Promotions
- Add Incentive Code and Apply Promotions

While the chapters and topics in this book are organized in the order you must configure the Apttus CPQ application, define fields, settings, and preferences to create an Incentive, you can skip to any section of the document that you want to know about.

To view the next page, click **Next Page** on the top right corner of the mid-section, select any topic listed in the **Table of Contents** displayed on the right side of the page. Every page in the Incentives Management section of the book also has links to other related topics, field descriptions, and procedures.

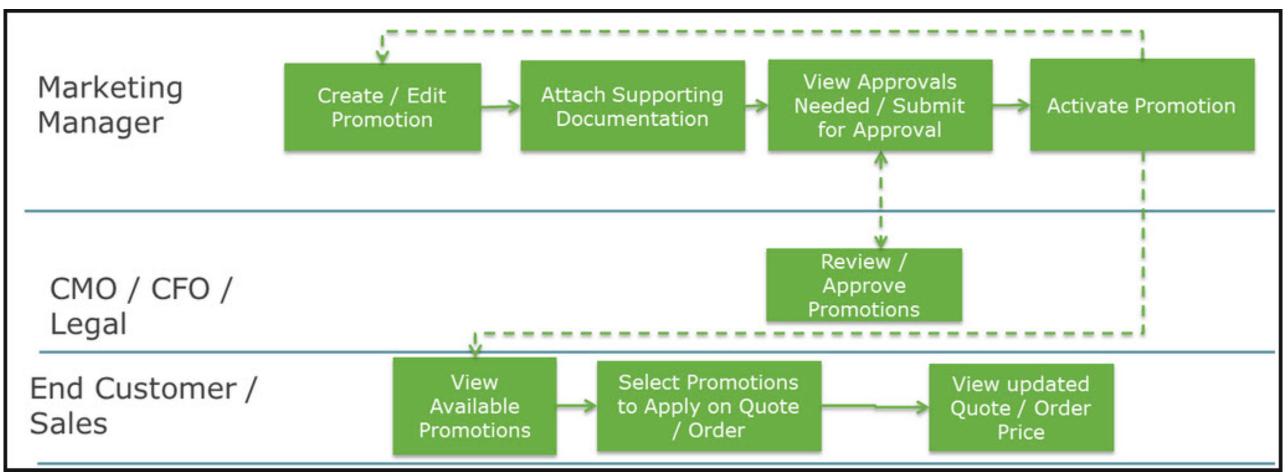
Click on any image in this document to view it in a carousel with a higher resolution.

If you are new to Promotions, let's get started. See the Promotions Workflow to understand the procedures and processes that go into creating a Promotion.

Promotions Workflow

For your organization to have a promotion event, the marketing team must first select the products to promote, define promotional pricing, and activate the promotion.

Promotional pricing may require approvals from your marketing manager and sometimes the finance and legal team.



Your sales representatives then can view the activated and available promotion to apply it on a quote or an order.

You can select from one or more different types of promotions that you want to create.

Types of Promotions

If you are a Marketing manager, you use historical transaction and sales data to create promotions for any one, or all of the following reasons.

- Increase Sales of a specific product
- Sell excess Inventory
- Clear stocks of product that is about to reach end-of-life
- Competitive Pricing
- Increase penetration in a given market or segment

In this release of the Apttus Promotions Management application, you can define the following types of promotions

- **Buy X Get X** - With a Buy X Get X type of promotion, you can offer your customers a specific number of a product if they purchase a certain quantity of the same product.

For example, you are a marketing manager in a telecommunications company. You are required to promote the sales of the data plans on offer. You can create a Promotion of type Buy X Get X where you offer your customers 1 GB of data free with every 10 GB that they buy.

- **For every X Get Y**- With a For every X Get Y type of promotion, you can offer your customers a specific number of a product if they purchase a certain quantity of a different product.

For example, you are a marketing manager for a software company. You are required to promote the sales of your latest designing software suite. You can create a Promotion of type For every X Get Y where you offer your customers a 1 year license and unlimited access to your style sheet templates with every 10 licenses that they buy.

Every Promotion you create has 4 aspects that you must define.

1. Promotion Information
2. Promotion Scope
3. Promotion Criteria
4. Promotion Limit

Each aspect of a promotion is dependent on certain Objects and fields that you must configure or define before you can create your first promotion.

You must complete a list of prerequisite tasks before you are ready to create your first promotion plan.

Configuring Promotions

With the Promotions Management application, you can now manage, execute, and analyze promotions using the Apttus CPQ product line. With the Promotions Management application, marketing managers can create new promotions, get internal approvals for such promotions, and roll these promotions to their sales channels.

You can use the Apttus Promotions Management solution to create and manage promotions like,

- buy X get X promotions
- tiered promotions
- automatically applied promotions
- coupon-based promotions

To understand how a promotion works, see the Promotions Workflow. Before you start creating a new promotion, see all the available Types of Promotions.

Prerequisites

If you are an Apttus CPQ administrator, you must first define who can access the Incentives Configuration page and what actions they can perform on this page.

You must create New Users and define permissions for each user of each profile as listed in the following table.

User	Profile	Permissions
Marketing Manager	• Create / Edit Incentive Programs • Submit Incentives for Approval • Analyse Incentive Impact of Quotes / Orders	Read / Write access on all objects used for Incentive Management Read access on Adjustment Line Items, Limit Data table, Rollup Data and other related object.

User	Profile	Permissions
	• Analyse Incentive Effectiveness	
Marketing Executives	• Analyse Incentive Impact of Quotes / Orders • Analyse Incentive Effectiveness	View Access on Incentive Objects, Quotes, and Proposals Read access on Adjustment Line Items, Limit Data table, Rollup Data and other related objects
Incentive Approvers	• Approve / Reject Incentives • View / Analyse Incentives	Read / Write Access of Incentive Object View Access on All objects used Incentives
Sales Representatives	Using Incentives on Quotes • View Available Pull Incentives • Add Incentive Code and Apply Incentive • View Incentive Benefit	View Access on Incentive Objects, Quotes, and Proposals, Adjustment line item objects, Limit Data
Customers	Using Incentives on the E-Commerce Orders • View Available Pull Incentives • Add Incentive Code and Apply Incentive	View Access on Incentive Objects, Quotes, and Proposals, Adjustment line item objects, Limit Data

User	Profile	Permissions
	• View Incentive Benefit	

For more information of User Profiles and permissions, see Profile Settings and Security.

If you are a Marketing Manager, you will want to use as much data as you can get to strategize and create an effective and efficient Incentive plan that guarantees meeting every business requirement. While you are already aware of several strategies to create an effective Incentive, you will use data and information such as

- Price Dimensions for a Promotion
- Formula fields for lookup objects that you can reference in a Promotion
- Child filters for the Formula fields you define
- Approval Workflows for a Promotion
- Groups of Incentives
- Create data roll-up

This data must already be available before you can implement or roll-out your Promotion. Let us see how you can access and then include or reference this data in the Incentives that you create.

Defining User Profiles and Permissions

You must create New users and define permissions for each user of each profile as listed in the following table.

User	Profile	Permissions
Marketing Manager	• Create / Edit Promotion Programs • Submit Promotions for Approval • Analyze Promotion Impact of Quotes / Orders	• Read / Write access on all objects used for Incentive Management • Read access on Adjustment Line Items, Limit Data table, Rollup Data and other related objects

User	Profile	Permissions
	• Analyze Promotion Effectiveness	
Marketing Executives	• Analyze Promotion Impact of Quotes / Orders • Analyze Promotion Effectiveness	• View Access on Incentive Objects, Quotes, and Proposals • Read access on Adjustment Line Items, Limit Data table, Rollup Data and other related objects
Promotion Approvers	• Approve / Reject Promotions • View / Analyse Promotions	• Read / Write Access of Incentive Object • View Access on All objects used Promotions
Sales Reps	using promotions on Quotes • View Available Pull Promotions • Add Promotion Code and Apply Promotion • View Promotion Benefit	View Access on Incentive Objects, Quotes, and Proposals, Adjustment line item objects, Limit Data
Customers	Using promotions on the E-Commerce Orders • View Available Pull Promotions • Add Promotion Code and Apply Promotion	View Access on Incentive Objects, Quotes, and Proposals, Adjustment line item objects, Limit Data

User	Profile	Permissions
	• View Promotion Benefit	

For more information of User Profiles and permissions, see Profile Settings and Security.

Configuring Incentive Custom Settings

With the help of custom settings, you have the ability to show or hide section and fields of Incentives that are not applicable for your business.

Hide or Show sections in an Incentive record

For sections displayed in an Incentive record and more configurations, configure the following custom setting:

1. Go to **Setup > Build > Develop > Custom Setting > Incentive Custom Display Setting**.
2. Click Manage next to **Incentive Custom Display Setting**.
3. Click **Edit** next to Promotions record. If there is no existing record, click **New**.
4. Enter the requisite details as follows:

Field	Description
Name	The mandatory name of the custom setting record. Type *Promotions* here.
Navigation Pages	The names of the sections which will appear in an incentive record on the left pane of the page. The sequence and name of the sections are defined here. The Information section is defaulted for every Incentive record. S: This is for the Scope section. C: This is for the Criteria section. B: This is for the Benefit section. D: This is for the Deduction section. L: This is for the Limit section. P: This is for the Coupons section.

Field	Description
	For example, if you specify *SCBL*, only the Scope, Criteria, Benefit, and Limit section will be displayed in the given order.
Show Exclusion Criteria	This check box indicates if you want to include Exclusion Criteria along with the default Inclusion Criteria under Criteria section for an incentive record.
Application Type	The name of the incentive type where this custom setting record will be applied. Valid values are: *Promotion*, *Rebate*, *Milestone Incentive.*

5. Click **Save**.

Configuring the fields displayed in an Incentive record

You can also control which standard and custom fields will be visible under different sections on your Incentive record using the following configuration.

1. Go to **Setup > Build > Create > Objects > Incentive > Field Sets**.
2. Click **Edit** next to the name of the fieldset for which you want to modify the fields.
   For example, if you want to edit the fields displayed in the Information Section, click Edit next to **Information Section Fields**. Search for your field, add or remove it from the fieldset.
3. Click **Save**.

Creating Promotion Groups

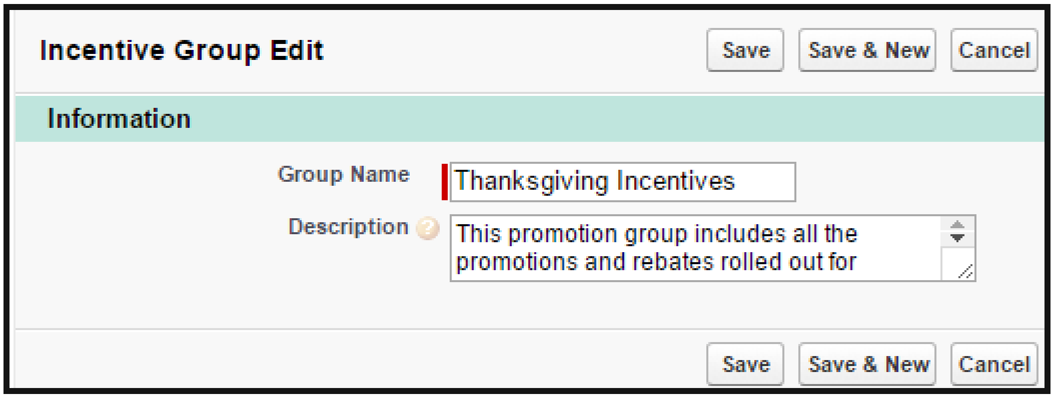
To analyze, report, and measure promotions, you can club together such promotions in categories called Incentive Groups.

One incentive, promotion or rebate, can be a part of multiple incentive groups.

To create a new incentive group,

1. Click **Incentive Groups** > **New**.
2. Enter a unique name for the incentive group.

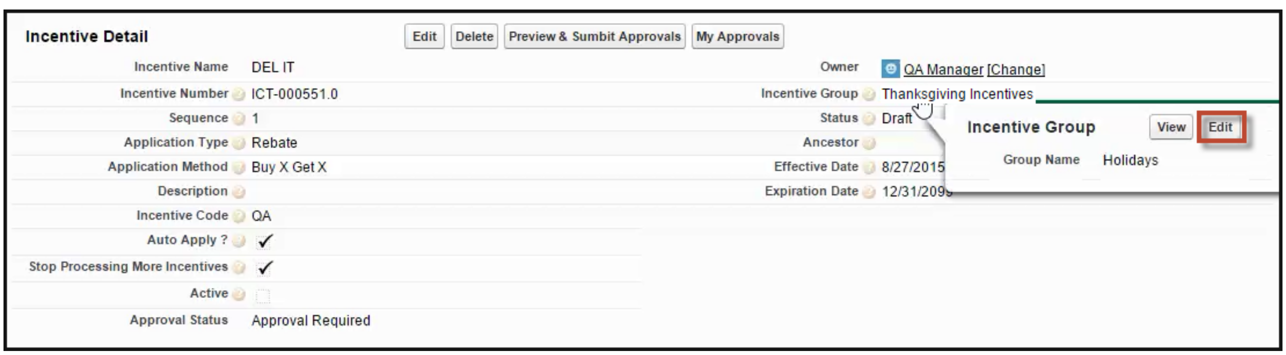
3. Describe the purpose of this group so others in your organization can use this group effectively.
4. Click **Save**.



5. In the Incentives section, click **New** to create an incentive in this new incentive group.

If you have already created your incentive and want to include it in an existing Incentive group, on the incentive detail page,

1. Click the Incentive Group field.



2. Click Edit.

3. Enter a unique name for the Incentive group.

4. Click **Save**.

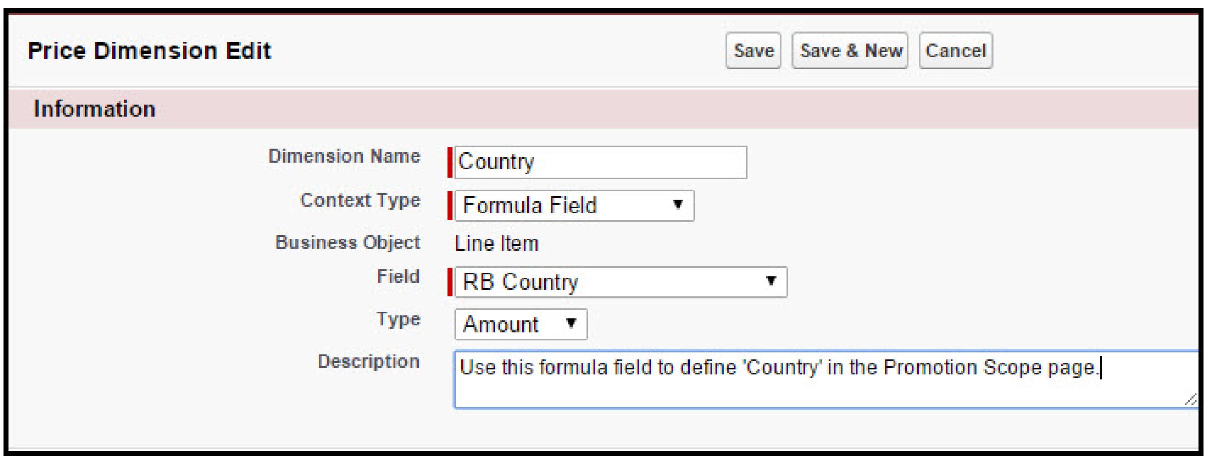
Creating a Price Dimension for a Promotion

When you define the scope of your promotion, you must include or exclude fields such as Region, Country, and Account, and Account Type, that impact the pricing of the purchase . You must define a Price Dimension of one or more of these fields in Salesforce that you want to include as a parameter to define the scope of your promotion.

For example, you are a marketing manager with an Online University. You want to roll out a promotion for prospective subscribers based only in the United States. Before you can define the Geographical scope of this promotion, you must create a Price Dimension for the **Country** object.

To create a price dimension for the Country object

1. From the force.com app menu, select Apptus Proposal Management.
2. Click Price Dimensions > New.



3. Enter the values for fields described in the following table.

Field	Description
Dimension Name	Enter a unique name for this Price Dimension.
Context Type	Select one of the following options. • Line Item - if this price dimension is only for the line item.

Field	Description
	• Formula Field - if this price dimension is based on the Field and Type you define below. • Child Filter - if this price dimension is based on filter criteria you define.
Business Object	Line Item (default)
Field	The field you select depends on the Context type. In this example, we selected Formula field and the objective is to create a price dimension for the Country field. The Country formula field for "Country" Price Dimension should point to the "CountryCode" field from the address as that is the field that has the pick-list associated with it.
Type	Select from one of the following options. • **Standard** - for a standard promotion • **Quantity** - if the scope of your promotion is based on quantity • **Term** - if the scope of your promotion is based on the term of the subscription • **Amount** - if the if the scope of your promotion is based on the purchase amount
Description	Describe the purpose of this price dimension so others in your organization can use it effectively.

4. Click **Save**.

For more information on Price Dimensions, see Creating Price Dimensions.

Next, you must create Promotion Groups to categorize the several promotions you configure.

Creating Formula Fields for lookup objects

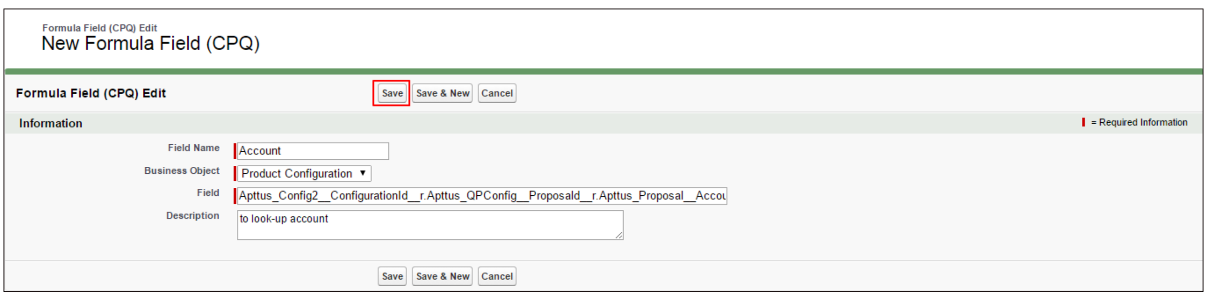
When you define the criteria of your promotions, you will require one or more lookup fields to fetch values from objects such as Quote, Order, Account, and Contact. For this purpose, you must create Formula fields that can reference the values you have defined for the fields in these objects.

> **Note**
>
> - Formula fields are not physical fields on the line item object.
> - Once you setup a new Formula Field, be sure to run the **Criteria Maintenance > Pricing Field Update.**

For example, you are a marketing manager who must define a promotion. You want to create and roll out a promotion that is applicable only to the healthcare industry. You must create a Formula field that references the Industry on the Criteria page. Here, we will create a formula field to create a look-up object.

To create a new Formula field

1. From the force.com app menu, select **Apttus CPQ**.
2. Click the **Formula fields** tab.



3. Enter values for the fields described in the following table.

Field	Description
Field Name	Enter a unique name for this new Formula Field.
Business Object	Select the business object you want to reference.

Field	Enter the API Name of the field that you want to look-up.
Description	Describe this formula field for the benefit of others in your organization.

4. Click **Save**.

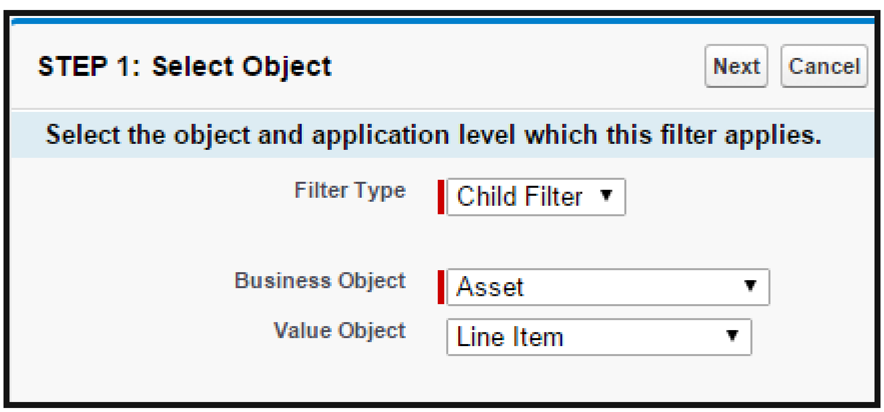
Creating Child Filters

When you define the criteria of you promotion, you will require reference to Order Line item, Asset Line Item, historical purchase summary, or other child objects of a look up object. These look-up objects can be Account, Contact, or User in the Promotion Criteria. You will need the **Search Filter (CPQ)** of type Child Filter. Using these child filters, you can also create the Price Dimensions to use in Benefit Matrix.

For example, you are a marketing manager who must define a promotion. You want to create and roll out a promotion that is applicable only to one specific asset. You must create a filter so you can reference the specific asset and map it to a field and specify a definite value.

To create a Search Filter,

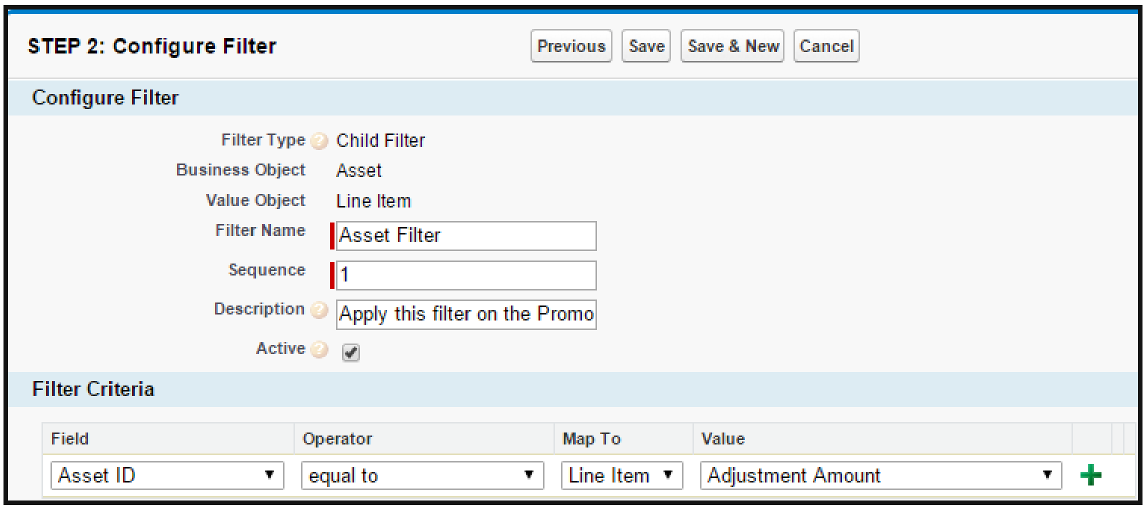
1. Click Search Filter (CPQ) > New.



2. Enter values for the fields described in the following table.

Field	Description
Filter Type	Select Child Filter.
Business Object	Select that Object you want to create a filter for.
Value Object	Select the object whose value you want to search for.

3. Click **Next**.



4. Enter values for the fields described in the following table.

Field	Description
Filter Name	Select Child Filter.
Sequence	Select that Object you want to create a filter for.
Description	Select the object whose value you want to search for.
Active	Select the check-box to activate this filter.
Filter Criteria	

Field	Description
Field	Select the field you want to reference.
Operator	Select the appropriate operator.
Map to	Select the field that you want to map your search filter to.
Value	Select the field whose value you want to populate in the promotions criteria.

5. Click **Save**.

Creating an Approvals Workflow for Promotions

You can define a hierarchy so that Promotions beyond a certain dollar amount or percentage require a manager's approval.

To create an Approval workflow for Promotions,

1. From the force.com app menu, select **Apptus Approvals Management**.
2. Click **Approval Processes**.
3. For the **Manage Approval Process for** field, select Incentive.
4. Click **Create new Process**.
5. Enter values for the fields described in the following table.

Field	Description
Process Name	Enter a unique name to identify this specific Approval Process.
Description	Describe this process for others in your organization.
Entry Criteria	
Field	Select the field that you want to apply the approval trigger for.

Field	Description
Operator	Select the appropriate option.
Value	Enter the value that must trigger the request for approval.

6. Click **Next**.
7. According to your Business requirements, select any one or more of the following check boxes.
    - **Consolidate Approvals**
    - **Consolidate Notifications**
    - **Continue Pending approvals on a Reject**
8. Assign an Email Template.
9. Assign an Approvals page.
10. Assign a Back-up Admin user.
11. Click **Save**.

For more information Approval Workflows, see Approval Functional End User Workflow.

Understanding a Data Rollup

If you are a Marketing Manager, you will require historical purchase data of customers and what products they have purchased. This information helps you effectively strategise and plan specific details for your promotion plans.

The purpose of a data roll-up is to convert different categories of historical purchase information into variables. You can use a data roll-up to generate a result that aggregates a hierarchy of values for one or more parameters.

For example, you are a Marketing Manager for a global electronics and consumer durables company. One of the products you want to promote is televisions. You must consider purchase history patterns of your customers from different geographical locations before you begin to strategize your promotion plan for each location. The purchase, pricing, and applied discount patterns for TVs varies greatly from one region to another. Notably, customers in the United States are more inclined to purchase TVs and other consumer durables for Thanksgiving which is in

November. Your customers in China on the other hand, are inclined to make such purchases during the Chinese New Year which is celebrated in February. There are more subtle differences such as your customers' brand loyalties that you can learn about using a data roll-up of historical purchases.

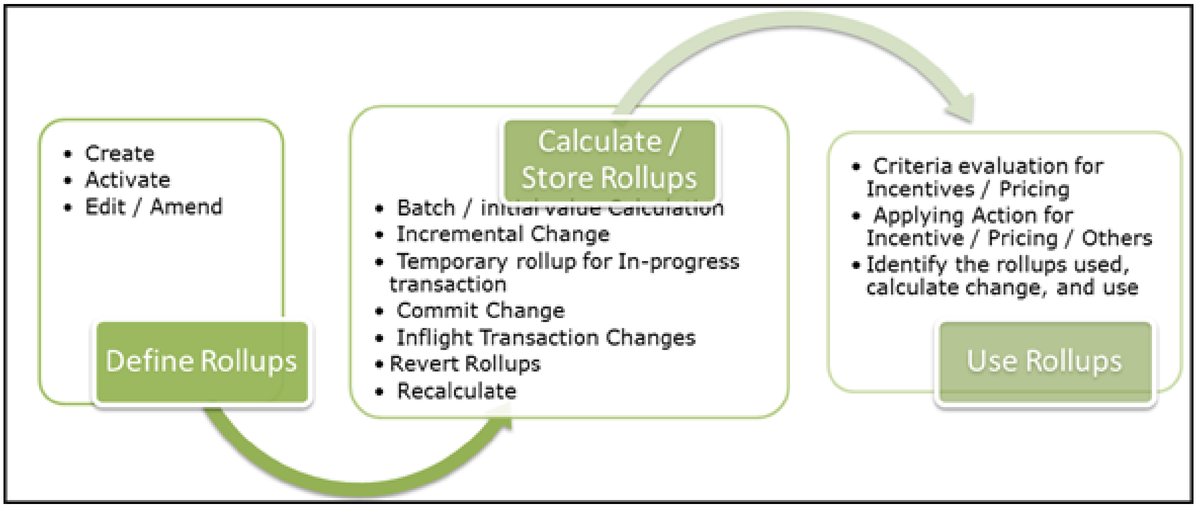
With Apttus Incentive Management, you can base your promotions on historical information such as how much your customers spent on specific products in a specific period.

The following table lists the factors you must consider and define to create a data roll-up.

Component	Description
Data Source	The **Order line items** and the **Cart line items** that your customers purchased are the source of all your historical purchase information.
Roll-up period	The period for which you want to measure the sales of a specific product or service. You must define this period with a specific Start Date and an End Date.
Roll-up function	A function is the operations you can perform with the data roll-up. At this time, **SUM** is the only function you can use.
Roll-up metrics	The metrics are factors that determine your customers' purchase behaviour. The two metrics you can use are • **Quantity** – the number of each product sold. You can use the Quantity metric to plan for inventory stocks, forecast projected sales and strategise an effective promotion. • **Amount** – the amount for which you sold each product. You can also use the Amount metric to measure the revenue you earned from each product sale and forecast revenues for the new promotion plan you want to define.
Roll-up criteria	The criteria the basis on which you roll-up historical purchase data. You can use data directly or reference it from the **Account**, **Product**, and **Contact** fields or their child records.
Group by	A groups is the parameter that defines how you categorize information in your data roll-up.

Roll-up scope	The scope defines the limits of your data roll-up. The scope is a mechanism to filter the list of the data roll-up that you must evaluate for a specific order.

Data Rollup Workflow

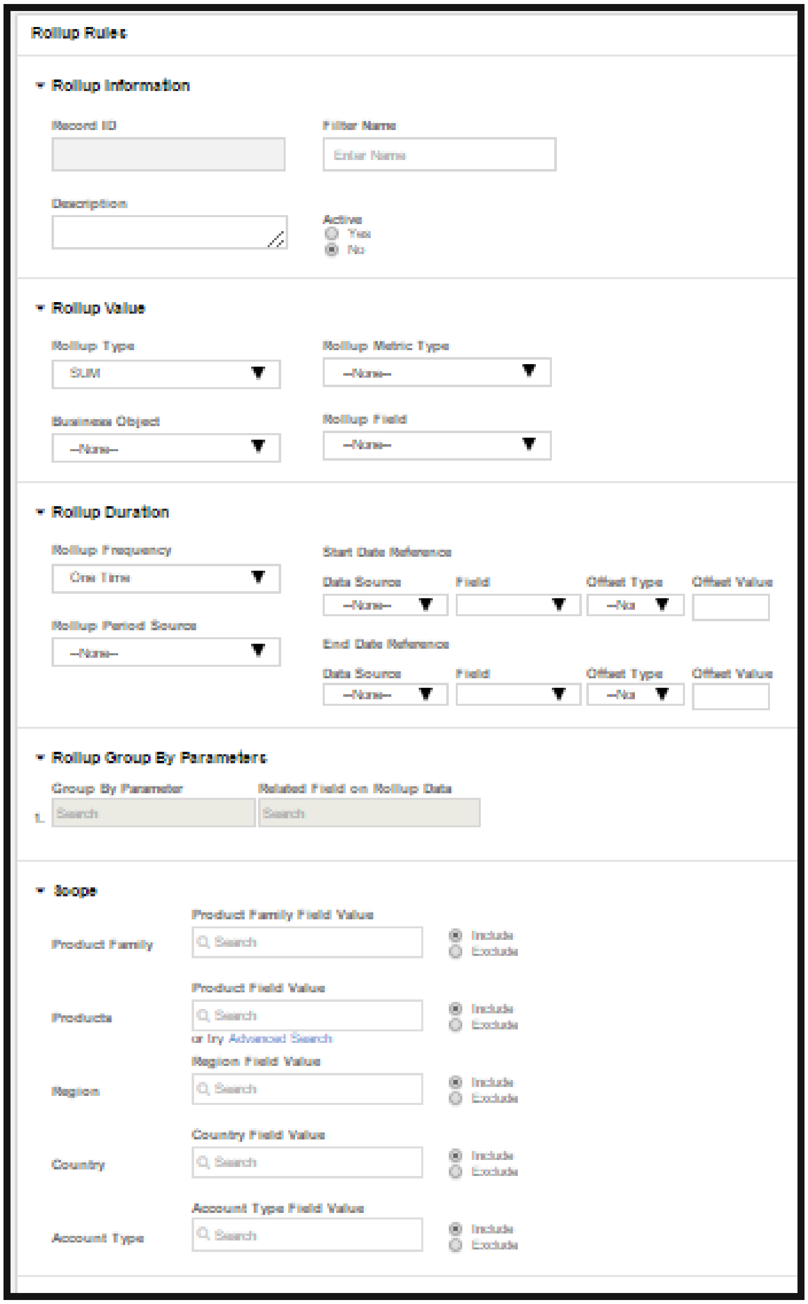


You must create a data roll-up to define the criteria, benefits, and the dimensions of a promotion plan.

Creating a Data Rollup

To create a new data roll-up,

1. Click **Incentive Rollups** > **New**.



2. Enter values for the fields described in the following table.

Field	Description
Rollup Information	
Record ID	

	This field is greyed out and Apttus Incentive Management automatically assigns this record an ID when you enter values for the remaining fields described in this table.
Filter Name	Enter a unique name for this filter. Consult your administrator for the naming conventions you must follow.
Description	For others in your organization to learn and use this data roll-up, describe the purpose and function of this data roll-up.
Active	Select the **Yes** radio button to activate this data roll-up and make it available for reference in your Promotions.
Rollup Value	
Rollup Type	Select **SUM** from the drop-down menu.
Rollup Metric Type	Select from one of the following options. • **Quantity** - the number of each product sold. • **Amount**- the amount for which you sold each product.
Business Object	Select from one of the following options. • Asset • Asset Line Item • Product • Product Attribute Value • Product Configuration
Rollup Field	Select from one of the following options. • Order Line Item • Cart Line Item
Rollup Duration	

<table>
<tr><td>Rollup Frequency</td><td>Select from one of the following options.
One Time – for purchases that require only a one-time payment
Recurring – for purchases that are subscriptions or usage-based.</td></tr>
<tr><td>Rollup period source</td><td>The source of the period depends is the Business Object you defined in the Rollup Value section.</td></tr>
<tr><td>Start Date Reference</td><td>Enter values for the fields described in the following table.
<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Data Source</td><td>The source of the data you want to roll-up is the Order Line Item or a child filter for which the value you define must be the Order Line Item.</td></tr>
<tr><td>Field</td><td>The field is the parameter that determines the period for which you want to generate the data roll-up.
Select from one of the following options.
Created Date
Effective Date
Expiration Date
Last Modified Date
Last Referenced Date
Last Viewed Date
System Modstamp</td></tr>
<tr><td>Offset Type</td><td>Offset values that you define determine the start date of the period for which you are creating a data roll-up.
Select from one of the following options.
Offset (+) Days
Offset (-) Days</td></tr>
</table></td></tr>
</table>

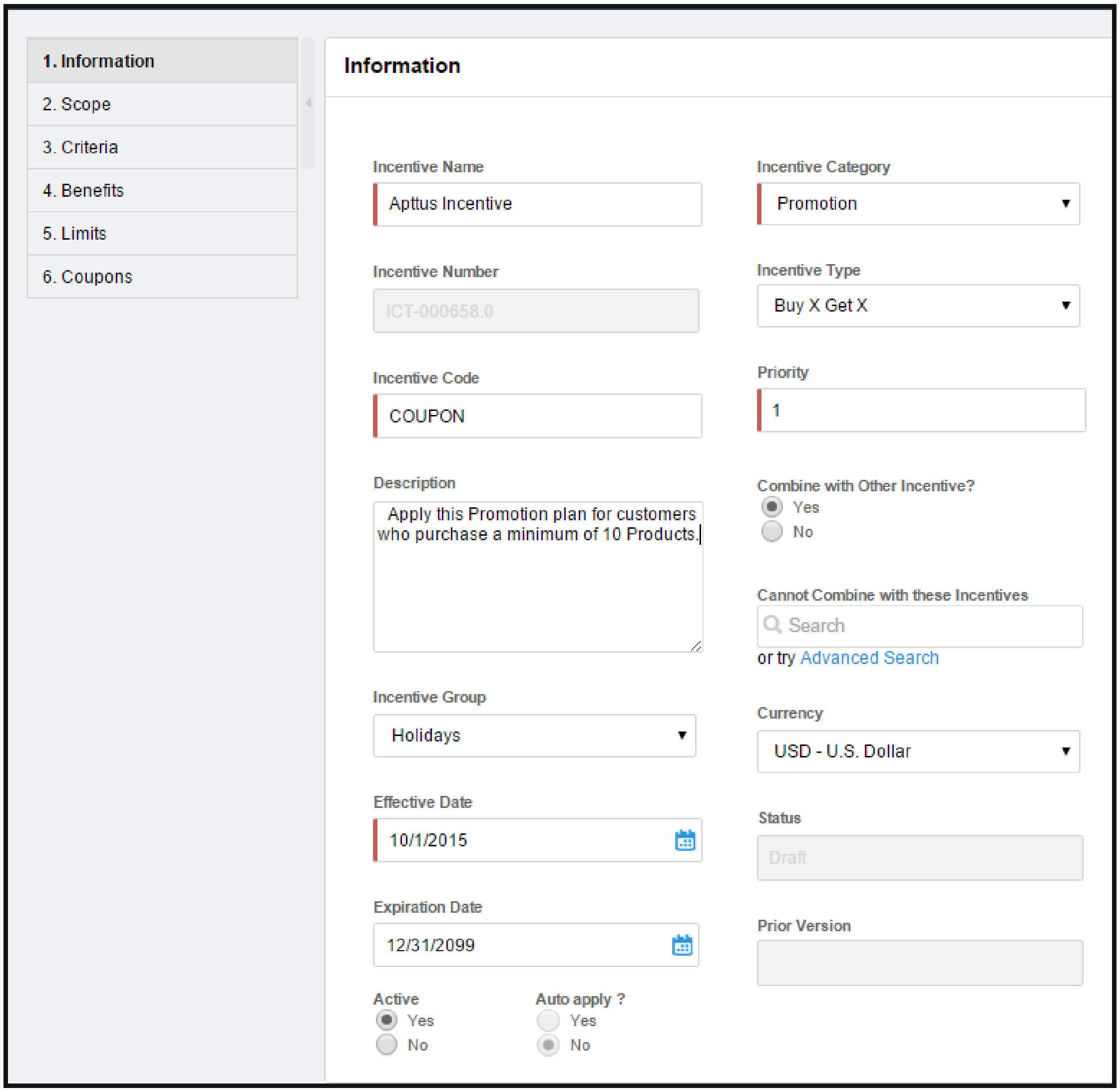
<table>
<tr><td></td><td>
<table>
<tr><td></td><td>
Offset (+) Months
Offset (-) Months
By default, the Value is constant.</td></tr>
<tr><td>Offset Value</td><td>Enter the number of days or months by which you want to offset the start date of the data roll-up.</td></tr>
</table>
</td></tr>
<tr><td>End Date Reference</td><td>Enter values for the fields described in the following table.
<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Data Source</td><td>The source of the data you want to roll-up is the Order Line Item or a child filter for which the value you define must be the Order Line Item.</td></tr>
<tr><td>Field</td><td>Select from one of the following options.
Created Date
Effective Date
Expiration Date
Last Modified Date
Last Referenced Date
Last Viewed Date
System Modstamp</td></tr>
<tr><td>Offset Type</td><td>Offset values that you define determine the end date of the period for which you are creating a data roll-up.
Select from one of the following options.
Offset (+) Days
Offset (-) Days
Offset (+) Months
Offset (-) Months</td></tr>
</table>
</td></tr>
</table>

	By default, the Value is constant.
	Offset Value — Enter the number of days or months by which you want to offset the end date of the data roll-up.
Rollup Group by Parameters	
Group by Parameter	This parameter defines how you will categorize information from the data source in your data roll-up. Lookup and search the field that you want to select.
Related field on Rollup Data	Lookup and search the related field that you want to select for your data roll-up.
Scope	
Note: To include the following fields in the scope of your data roll-up, select the **Include** Radio button.	
Product Family	Lookup and search for the product families that you want to include or exclude from the scope of your data roll-up.
Products	Lookup and search for the products that you want to include or exclude from the scope of your data roll-up.
Regions	Lookup and search for the regions that you want to include or exclude from the scope of your data roll-up.
Country	Lookup and search for the countries that you want to include or exclude from the scope of your data roll-up.
Account Type	Lookup and search for the Account Types that you want to include or exclude from the scope of your data roll-up.
Rollup Criteria	

<table>
<tr><td>Data Source</td><td><table><tr><th>Left Side</th><th>Right Side</th></tr><tr><td>Select Order Line Item from the drop-down list.</td><td>Select Order Line Item from the drop-down list.</td></tr></table></td></tr>
<tr><td>Field</td><td><table><tr><th>Left Side</th><th>Right Side</th></tr><tr><td>Select a field from the Order Line Item object.</td><td>Select a field from the Order Line Item object.</td></tr></table></td></tr>
<tr><td>Operator</td><td>Select from one of the following options.
<
>
<=
>=
=</td></tr>
<tr><td>Value Type</td><td>Select from one of the following options.
Offset (+) Days
Offset (-) Days
Offset (+) Months
Offset (-) Months</td></tr>
<tr><td>Value</td><td>Enter the number of days or months by which you want to offset the end date of the data roll-up.</td></tr>
<tr><td>Criteria Expression</td><td>Apttus Incentive Management populates this field when you enter values for the fields that defines the scope.</td></tr>
</table>

3. Click **Save**.

Defining a Promotion



To define a new promotion,

1. Go to the Salesforce App Menu and select **Apttus Incentive Setup**.
2. Click **Incentives** > **New**.
3. On the Incentives information page, define values for the fields described in the following table.

Field	Description
Incentive Name	User specific name for the Incentive. This promotion name is displayed to the user at any place the Incentive information is displayed. For example, whenever a user looks-up the promotion information applicable to a specific product, the promotion name you provide here is displayed.
Incentive Number	System generated unique number for the Incentive. This differs from the Salesforce record id. When an active incentive is changed over time, the Incentive number remains same.
Incentive Code	This is the code that is used for applying incentive. This code is captured on the order or proposal line items when an incentive is applied to the line item. By default, this value is set to Incentive Number, but is editable. This can also be used by the marketing team for marketing campaigns. When a promotion is not auto- applied, that is when the **Auto Apply ?** check box is cleared, the user can use the incentive code to avail a promotional offer.
Description	User entered description for the incentive.
Incentive Group	This is a generic grouping of the Incentives for analyzing and tracking the incentives. User can select an Incentive group for each incentive. The list of Incentive Groups needs to be defined separately in the Incentive Group object and should be active. Marketing managers can create reports and track the incentive groups that are more successful than others and drive the business strategy accordingly.
Currency	You can define the currency for which this Promotion is applicable. Select from the list of currencies available. **Note** Consult your Apttus CPQ Administrator to enable the Advanced currency management feature for your org. You must make sure that you have also defined the currency denomination in the

Field	Description
	• Price Rule Set • Price Rule • Price Rule Entry
Start Date	Start Date from which the promotion is applicable.
Expiration Date	Date on which the promotion expires. Can be blank Needs to be greater than Start Date
Active?	Can be set to Yes or No. Set it to Yes when you want a promotion to be applied and active for the criteria you specify. Active flag on the promotions will be set using a workflow linked to approvals similar to what you would do for any other custom object approval. For example, you can write a workflow rule to set the active status to true once the promotion is Approved (if approval is setup). Approvals set up would be similar to custom object approvals for any object.
Auto Apply?	Can be set to Yes or No. Set it to Yes when you want a promotion to be automatically applied when a promotion criteria is met. Set it to No when you want the user to manually enter an Incentive Code to apply promotions.
Incentive Category	Promotion Rebate
Incentive Type	When you select the Incentive category as Promotion, the incentive type you can select is Buy X Get X.
Priority	Specifies the priority of the promotion. For example, if more than one promotion is applicable for a product, the promotion with a higher priority is applied.
Combine with Other Incentive?	Enables you to combine the existing incentive with another incentive.

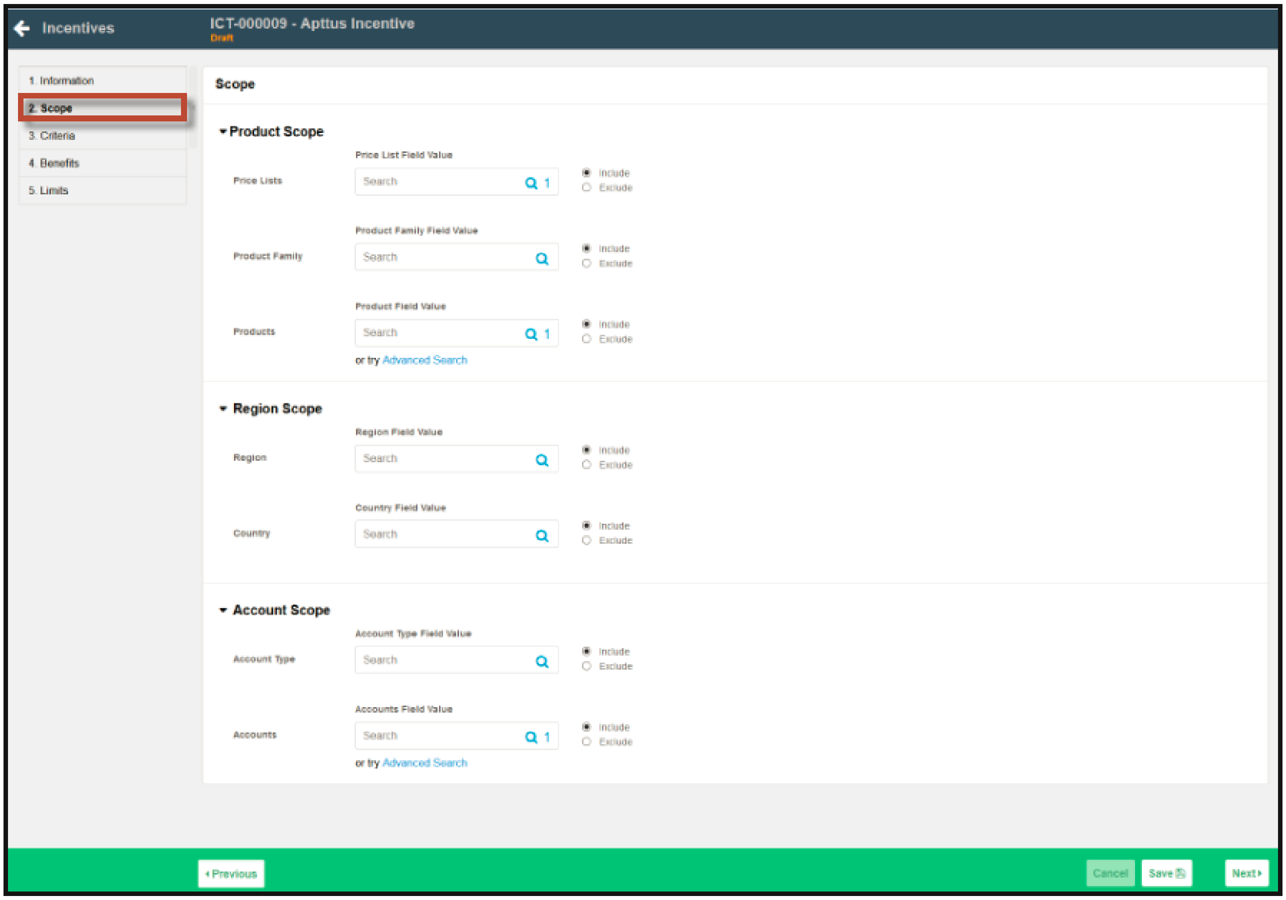
Field	Description
Cannot Combine with these Incentives	This is the lookup to other incentives that is stored in the child object - Related Incentives. The created incentive cannot be combined with the incentives you specify in this search field.
Status	Specifies the status of the Promotion. The status of the promotion is dependant on the promotion lifecycle. Once a promotion is approved, you can set the status of the approval as active. Approvals on Promotions - Admins can setup approval processes and other workflows on promotions. Marketing managers can send promotions for approval and approvers can view the promotion, approve or reject the promotions with or without comments, and activate it.
Prior Version	This field will show the lookup to the previous version of the incentive. User should be able to click and open the previous version using this link

4. Click **Save**.

After you have described the promotion, you must now define the scope of this new promotion.

Defining the Scope of a Promotion

You must define the scope of a promotion to determine the types of accounts, the regions, and the products that this promotion will be available to.



To define the scope of your new promotion,

1. On the left side work pane, click **Scope**.
2. Define values for the fields described in the following table.

Note

The default value for each field is **All**.

By default, the fields you define are set to be included in the scope of your promotion. Select the **Exclude** radio button for any parameter if you want to exclude that specific object from the scope of your promotion.

Field	Description
Price List	Select the price list that you want to use for your promotion.

Field	Description
Product Family	Select the Product Family that you want to apply the promotion to.
Products	Select the specific products that you want to apply the promotion to.
Region	Select a geographical regions of the customers that you want to make this promotion available to.
Country	Select a countries where you want to make this promotion available.
Account Type	Select the types of accounts for which you want to make this promotion available.
Account	Select the accounts for which you want to make this promotion available.

3. Click **Save**.

After the scope you must now define the criteria for this promotion.

Defining the Criteria of a Promotion

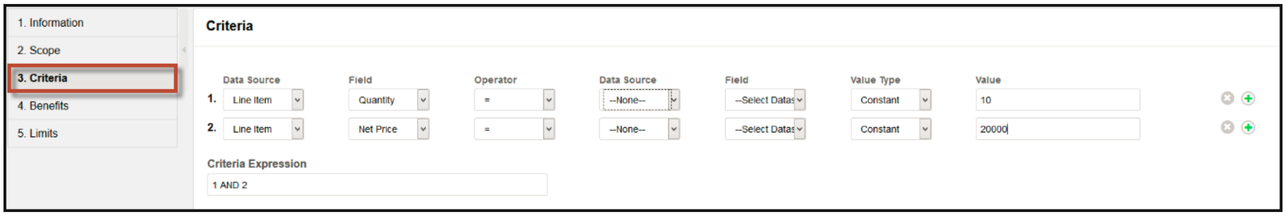
You must define the criteria of the promotion before you are ready to make it available to customers. If a sales representative wants to apply this promotion to a customer's purchase, the criteria you define here will determine if that purchase qualifies for the promotion.

You can define two types of criteria,

1. based on any the past purchases of customers or rolled up metrics of past purchases.
2. based on data from the line item, header, lookup objects, and child objects of the lookup objects for a quote or an order.

You can also define additional filter criteria to determine which products this promotion is applicable for.

The Line Item Object and the Roll-up Data Object are the data sources for the Additional Filter Criteria .



You can also define a Data source to compare the Line Item to an Incentive Roll-up field. For example, the promotion you create can be applied to the cart where the Line Item Total Quantity > Roll-up field Quantity.

To define the scope of your new promotion,

1. On the left side work pane, click **Criteria**.
2. Define values for the fields described in the following table.

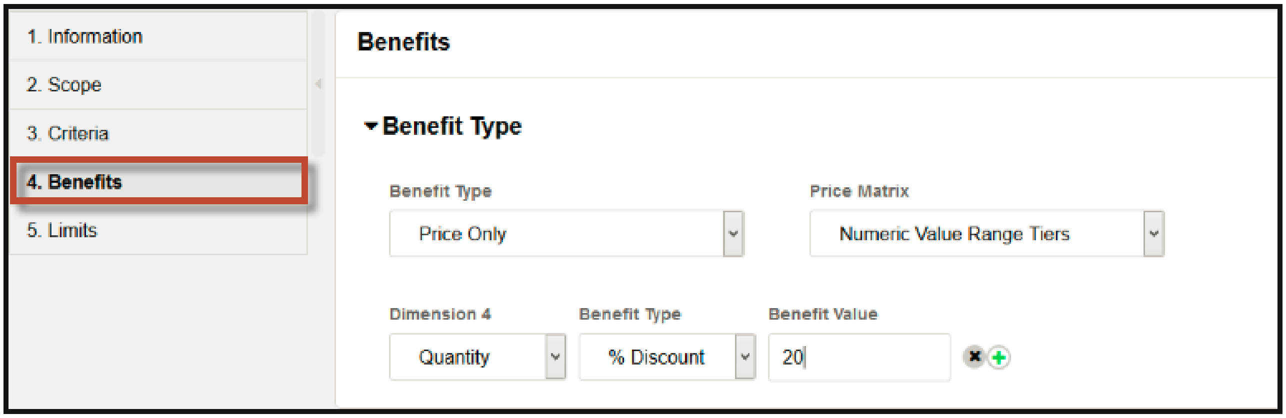
On the left side of the criteria page.	
Field	**Description**
Data Source	Select Line Item. This line item is the primary object to which your promotion is applied.
Field	Select one option from this list of fields and Apttus formula fields you have created for the line item that are available for the line item you selected as the data source.
Operator	If the field you defined is a pick-list or a lookup field, you can select one of the following operators. • = • != • IN • NOT IN For any other field you defined, select one of the following operators. • < • <=

On the left side of the criteria page.	
Field	**Description**
	• > • >= • = • !=
On the right side of the criteria page.	
Data Source	Select one from the list of Search Filter CPQ data sources linked to Line Item you selected as the primary data source. With this data source, you can use the information from child objects to create evaluation criteria that can be based on previous purchase history such as order or asset line items, or roll-up data.
Field	Select one from the list of fields from the child filter object.
Value type	While the default value is constant, if the field you selected is date, you can select from one of the following options. • Offset (+) Days • Offset (-) Days • Offset (+) Months • Offset (-) Months
Value	Enter a simple constant value or select single or multiple values using search widget depending on the field type and operator you defined for the primary data source.

3. Click **Save**.

You must now define the benefits of this promotion.

Defining the Benefits of a Promotion



To define the benefits of a promotion,

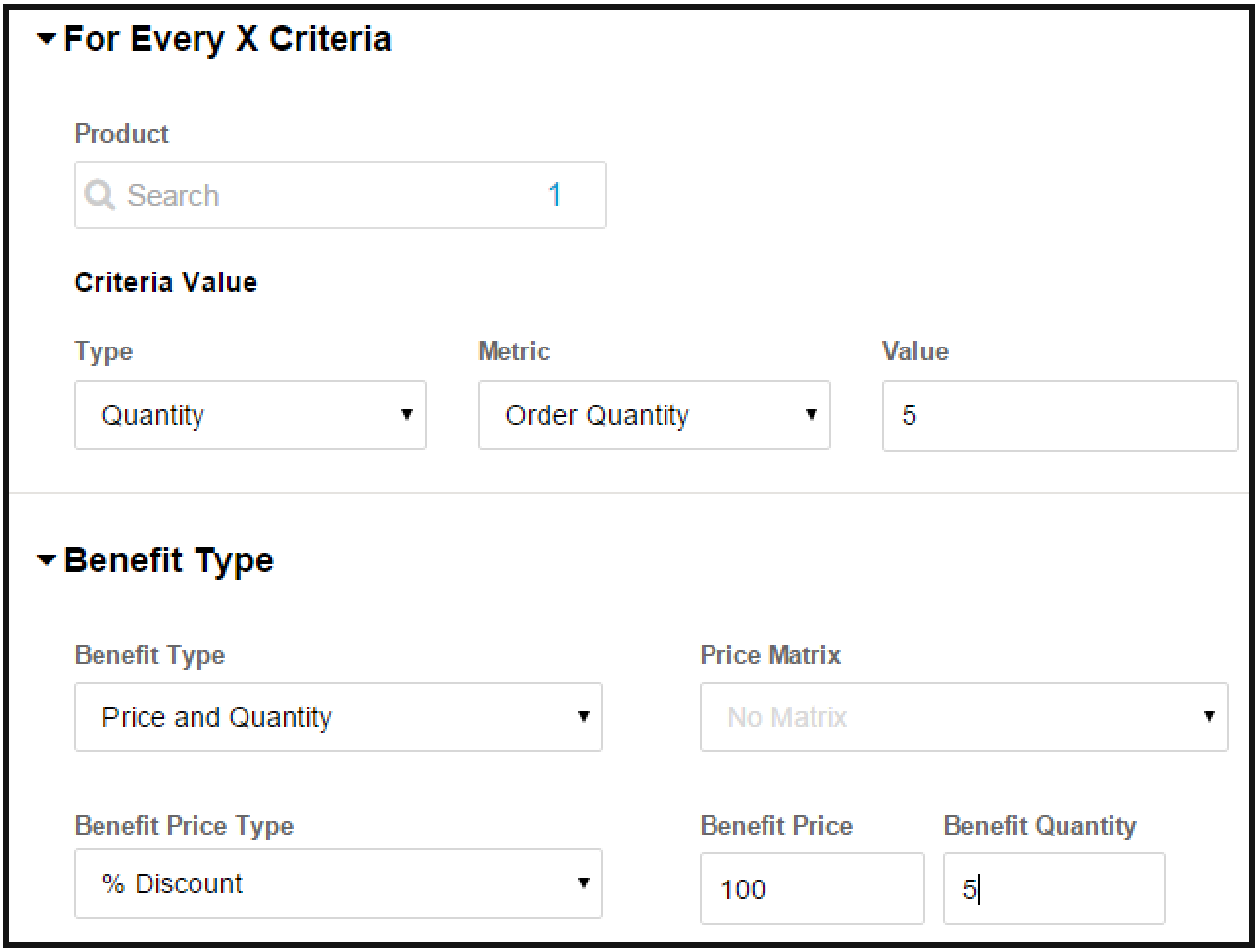
1. On the left side work pane, click **Benefits**.
2. Define values for the fields described in the following table.

Benefit Type	Description
Price Only	Enables you to apply a discount, markup, or override the price for the product. You can specify the price by which the product will be discounted, marked up, or overridden. For example if you set the Benefit Type as Price Only, set no matrices, set the Benefit Price Type as % Discount, and Benefit Price as 50. The price of the added products after filtration will be discounted by 50%. You can also define benefit pricing tiers for specific duration.

3. Click **Save**.

For every Benefit Type you can define a Price Matrix. To know what a price dimension is and how to define price dimension ensure that you Create Price Dimensions . A benefit matrix enables you to define the incentives based on different values you specify. These are an advanced pricing concept used to define tiered pricing paradigms, or complex pricing structures with multiple criteria. Common examples are pricing tiers for a product based on user count or particular customer or transactional dimensions. The table below outlines the scenarios for each of the price matrices.

Price Matrix	Description
Discrete Value Tiers	Enables you to create a price matrix for a dimension with a discrete value. For example, for an account type silver provide a discount of 10%, for gold provide 20, and for platinum provide 30%.
Numeric Value Range Tiers	Enables you to create a price matrix for a dimension with a numeric value. For example, if the product quantity is 10 provide a discount of 10%. for 20 provide a discount of 20% and for 30 provide a quantity of 30%. For Numeric Dimensions it is imperative that you define the value in the text field given.
Discrete & Numeric Range Tiers	Enables you to define a price matrix for a dimension with both discrete and and numeric values. For example, for an account type silver, where product quantity is 10- provide a discount of 10%, for gold where product quantity is 20 provide a discount of 20%, and for platinum where product quantity is 10 provide a discount of 30%.



For every X get X type and for every X get Y type of Promotions, on the Benefits Page you must,

1. Select one or more products that you want to include benefits for and promote.
2. Define the Criteria. Enter values described in the following table.

Field	Description
Type	Select Quantity. This is only type of criteria available to define the promotions benefit.
Metric	Select any available matrix of type quantity that you have already defined.
Value	Enter the value for which the benefit of the promotion becomes applicable.

3. Define the Benefit type. Enter values described in the following table.

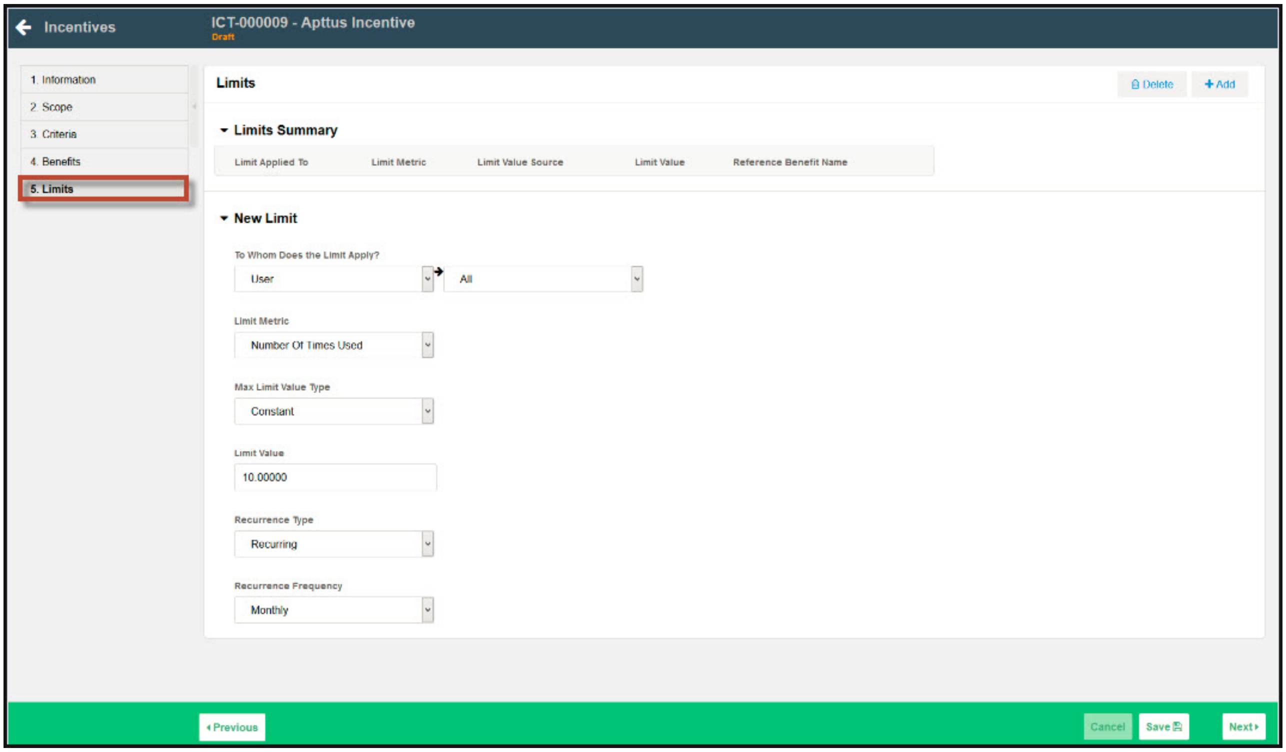
Field	Description
Benefit Type	Select Price and Quantity. This is only type of benefit type available to define the promotions benefit.
Price Matrix	
Benefit Price Type	Select from one of the following options. • %Discount • Discount Amount • %Markup • Markup Amount • Price Override • Price Factor
Benefit Price	Enter a number to define the Benefit Price Type.
Benefit Quantity	Enter a number to define the benefit your customer will get from this Promotion.

4. Click **Save**.

With the basic information, scope, criteria, and benefits defined, you must now define the limits of this promotion.

Defining the Limits of a Promotion

You must define promotion limits to determine the number of times you customer can apply a promotion for different types of the transactions.



To define the limit of a promotion,

1. On the left work pane, click **Limits**.
2. Click **Add** to add a new limit and **Delete** to delete a limit.
3. Enter values for the fields described in the following table.

Field Name	Description
To Whom Does the Limit Apply?	Enables you specify the object on which the limit is applied. The objects on which limit is applicable on are *Account*, *User*, and *Contact*.
Limit Metric	Specifies what will be the parameter on which the limit is applicable. *Benefit Quantity*: Specifies that the discount would be applied on a fixed quantity.

Field Name	Description
Max Limit Value Type	Specifies whether a limit would be a constant value or derived from some formula field.
Limit Value	Specifies the value of the Limit Metric. For example, if you specify the limit value as 10 and the limit metric as Benefit Quantity, the user can use the same promo code when the Quantity of the product is 10.
Recurrence Type	Specifies whether the limit applied is one time or on a recurring basis.
Recurrence Frequency	Specifies the frequency at which the applied incentives are limited. Available frequencies are *Monthly*, *Quarterly*, *Half Yearly*, *Yearly*, *Order*.

Multiple promotion Limits

- A given promotion can have more than one limits and all these limits needs to be enforced for the promotions. E.g. Promotion limits on a single promotion can be as follows.

Promotion Limit Parameters

- Overall Promotion Limit - Across customers
    - Irrespective of duration -Total number of promotions applied across customers.
    - By specific duration - Total number of promotions applied across customers during a given period.
- Limit by Customer
    - Irrespective of duration -Total number of promotions applied for a given customer.
    - By specific duration - Total number of promotions applied for a given customer during a given period.
- Limit by Customer and Product
    - Irrespective of duration -Total number of promotions applied for a given customer and product.

  - By specific duration - Total number of promotions applied for a given customer and product during a given period.
- Order level Limits: This means that customer can only use certain number of promotions within a given order.
  - By promotion
  - By required purchase product level
  - Order level promotion limits can be dynamically applied and does not need to be tracked separately. However, account level and promotion level needs to be tracked.

For example, all the eligible doctors within a DSO (Days Sales Outstanding) will receive 1 case (Invisalign Teen, Full or Assist) priced at $990, if that case is submitted within 90 days of their re-engagement training. Else, they are not eligible for receiving cases worth $990.

Expected Behavior

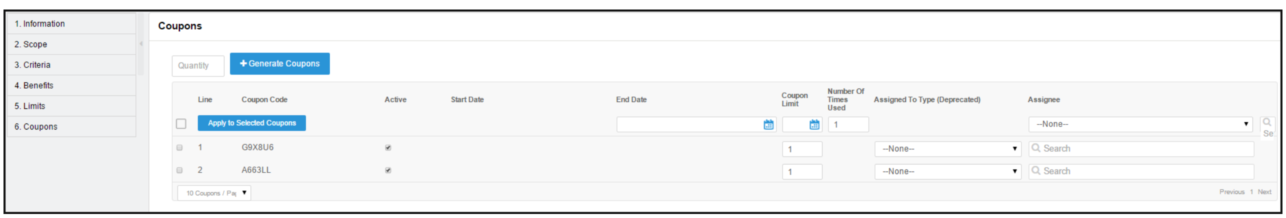
Options Selected	Expected behavior
Auto Apply deselected, Combine with Other incentives selected	You can build a User Interface to display the “Show Available Promotions” list using the Promotions API, which displays non-auto applied promotions. The user can then select one or multiple promotions to be applied.
Auto Apply deselected, combine with other incentives deselected	If a user provides multiple promo codes as a comma separated list, the highest priority promotion is selected and applied from the list of promotions provided in the comma separated list.
Auto-Apply selected, combine with other incentives deselected	If two or more promotions are applicable, the system picks the promotion with the highest priority and applies it.

Options Selected	Expected behavior
Auto-apply selected, combine with other incentives selected	All promo codes fulfilling the criteria will be applied. If one or more promo codes have priority 1 which is applicable in a scenario all the promotions are applied. Suppose for a given scenario there are 4 promo codes which are applicable having priority 1,1,2,2 all four are applied. Pricing is not stacked, so if all the four qualify and allow others to be applied then all four are applied. If some of the promotions do not allow others to be applied then those promotions are filtered out.

After defining promotions, ensure that you run **Criteria Maintenance > Update Pricing Fields** to apply the defined promotions.

Generating Coupons

Coupons are generally alphanumeric codes that correspond to a promotion.



To generate coupon codes for a promotion that you have created,

1. On the left work pane, click **Coupons**.
2. Enter the number of coupons you want to generate for this promotion.
3. Enter values for the fields described in the following table.

Field	Description
End Date	The last date this coupon is valid.
Number of Times Used	The number of times you can use this coupon.

Assignee	The assignee is the level at which the coupon is valid. Select from one of the following options. • User • Account • Contact

4. Click **Generate Coupons**.

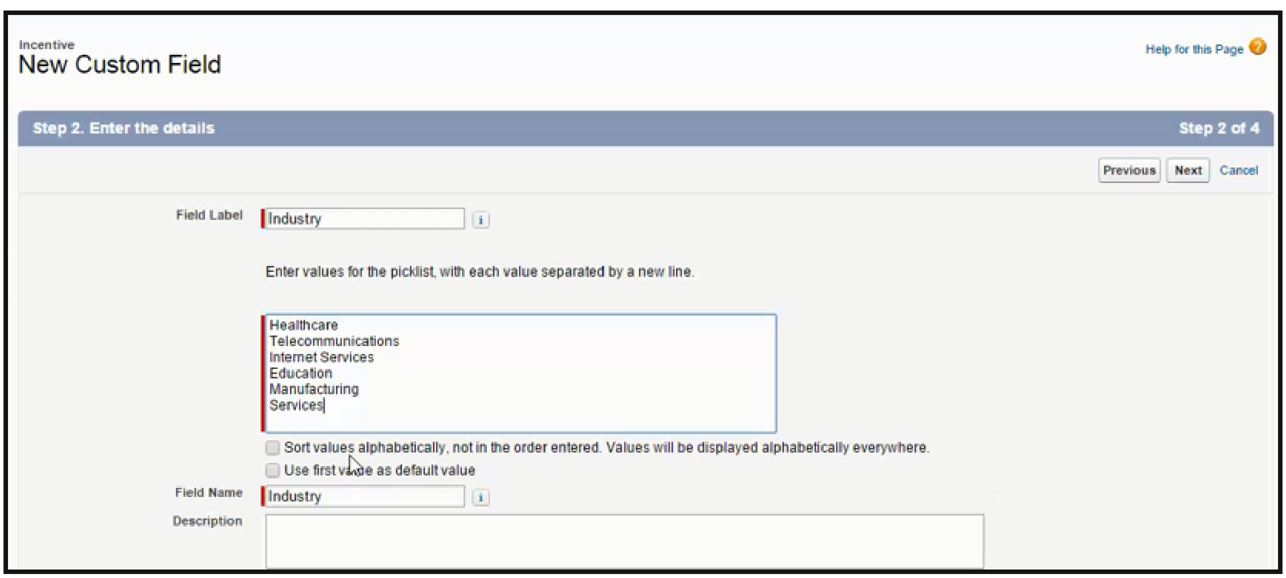
Adding Custom fields to the Promotion Information Page

You can now add to the details that you want to capture on the Promotion Information page. You can include more fields such as check boxes, number, dates, texts, currency, pick lists, multi-pick lists, and other fields that Salesforce supports.

For example, you want to include a field to capture the industry for which you are rolling out this specific promotion. You can add such a custom field on the Promotion Information page.

To add a custom field on the Promotion Information page

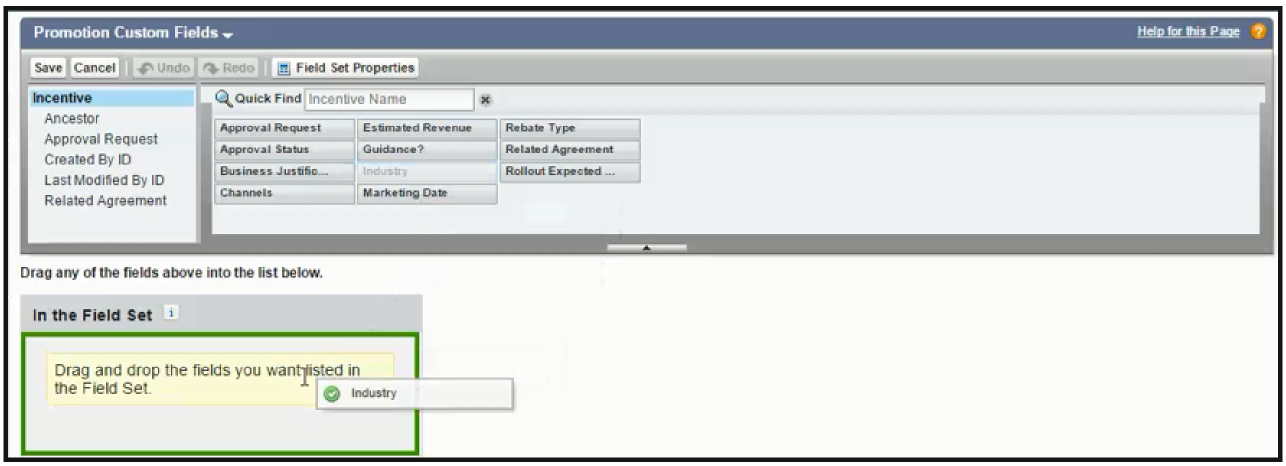
1. Add a custom field for the Incentive object. Let's call this field **Industry**, which is a pick list from which users can select a relevant option.



2. Add the pick list options you want to include. For example,

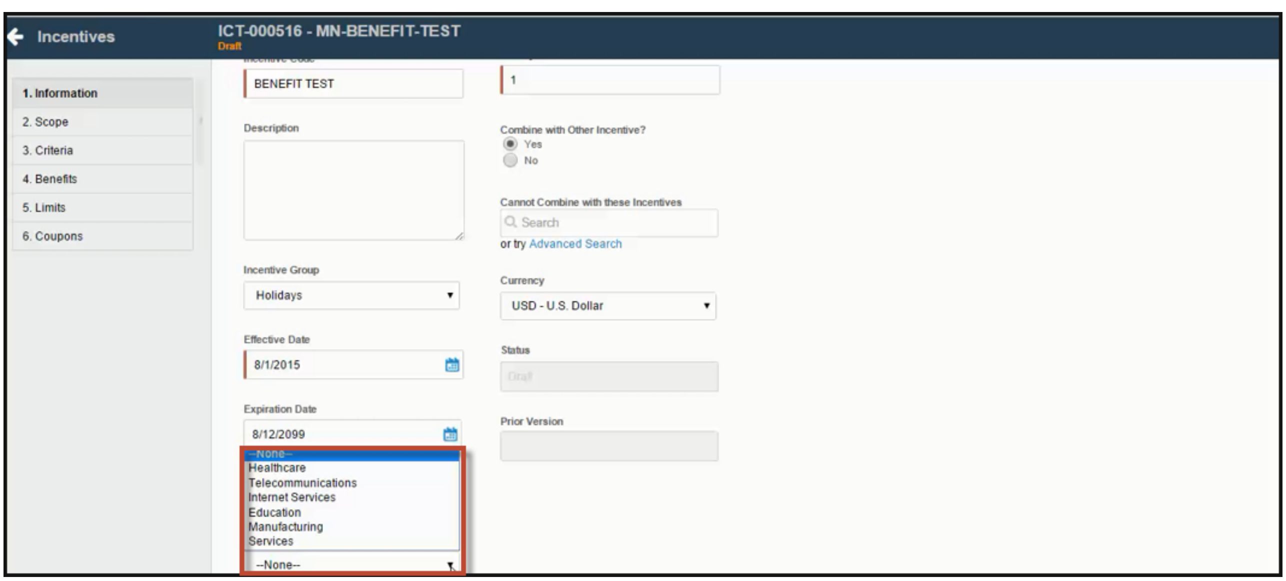
- Healthcare
- Industry
- Manufacturing
- Services
- Telecommunication
- Others

3. Now Edit the Incentive Field set to add the Industry field.



4. Click **Save**.

You have successfully added the Industry pick list to the Promotion Information page.



Managing Coupons

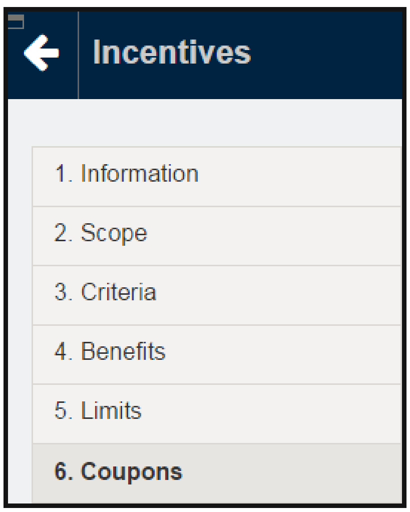
With Apttus Promotions Management, you can now create multiple coupon codes for potential customers to promote your business and specific products in your inventory.

After you identify potential end customers in a target market, you can use coupons to track how customers use incentives, how you can optimize your promotions, and how you can re-target your customers. You can generate multiple coupon codes to share the range with your marketing team.

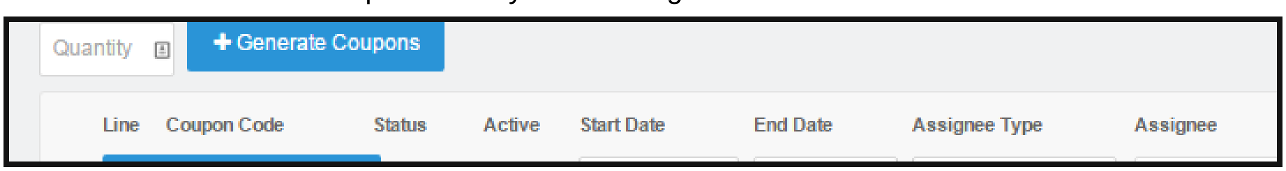
The Coupons tab is visible only if you defined the *Auto Apply = False* on Incentive Information page.

To create coupon codes,

1. On the incentives page, click **Coupons**.

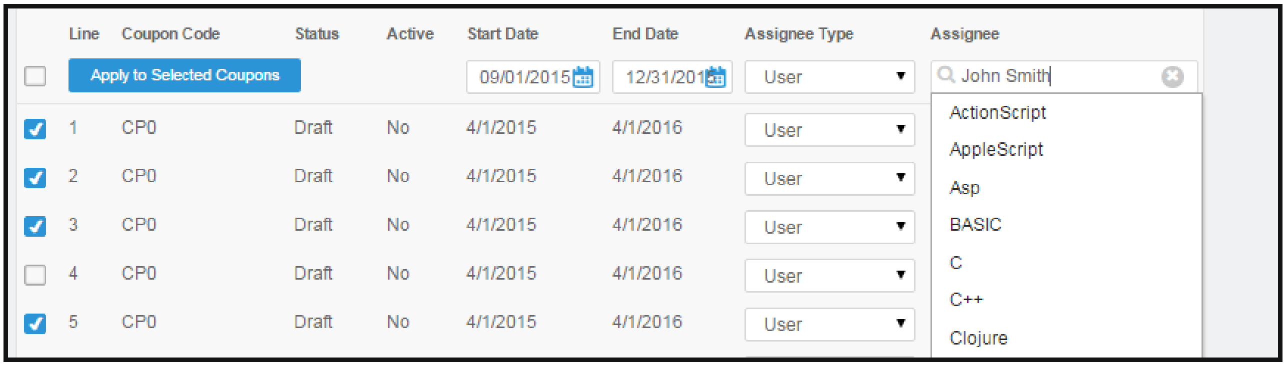


2. Define the number of coupon codes you want to generate.



3. Click Generate Coupons.

4. On the header, define the **Start Date**, **End Date**, **Assignee Type**, and the specific **Assignee**.

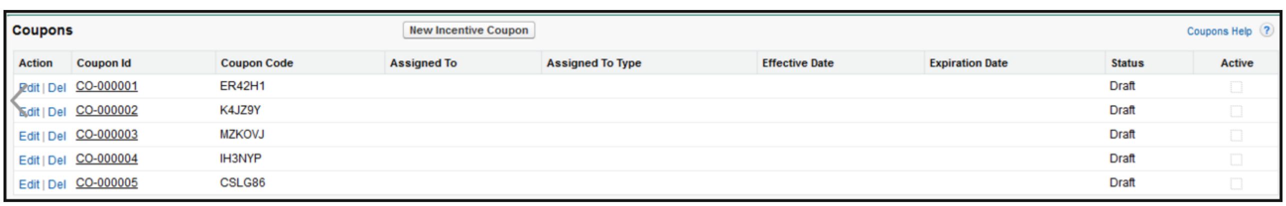


5. Select specific coupons in the list and click **Apply to Selected Coupons**. You can also edit coupons en mass, simply select the **Apply to Selected Coupons** before you click the button.

6. To create multiple coupons for one incentive, use the following sample code.

```
List<IncentiveCoupon__c> coupons = IncentiveCouponFactory.
createSObjectsForIncentive(incentiveSO.Id, 5);
            System.assertEquals(coupons.size(), 5);
            for (IncentiveCoupon__c couponSO : coupons) {
                  System.debug('couponCode=' + couponSO.
CouponCode__c);
            }
```

You can view the resulting list of coupons on the Accounts page.



Applying Promotions for eCommerce Orders

You can apply the promotions that you define, for orders placed on Apttus eCommerce, using APIs.

The following table lists describes such APIs.

API	Description
getIncentivesForCart	To get a list of Incentives of type PULL (Auto Apply = False) for a given Cart for an order.
updatePriceForCart	To apply one or more PULL promotion, set the comma separated list of PULL promotion code (Incentive codes) on the Product Configuration object in the field " Apttus_Config2__CouponCodes__c". After setting the values, reprice the cart using "updatePriceForCart" API. System will auto evaluate all promotions of type PUSH (auto Apply = True) and will apply qualified promotions as part of the pricing.
applyIncentiveLimit	API to apply commit incentive limit from work-in-progress orders to the final limit data. Ideally, this API will be called when order is accepted or confirmed by the customer. However, individual implementations can decide when to call this API.

Analyzing Promotions

For analysis purpose, Apttus CPQ and Promotions Management captures the information of the applied promotions at the line item level in the promotions code and promotion benefit. As a Marketing Manager, once the promotions are applied on the sold products, you can perform different types of analysis by creating reports and dashboards for the applied promotions and forecast your business revenue.

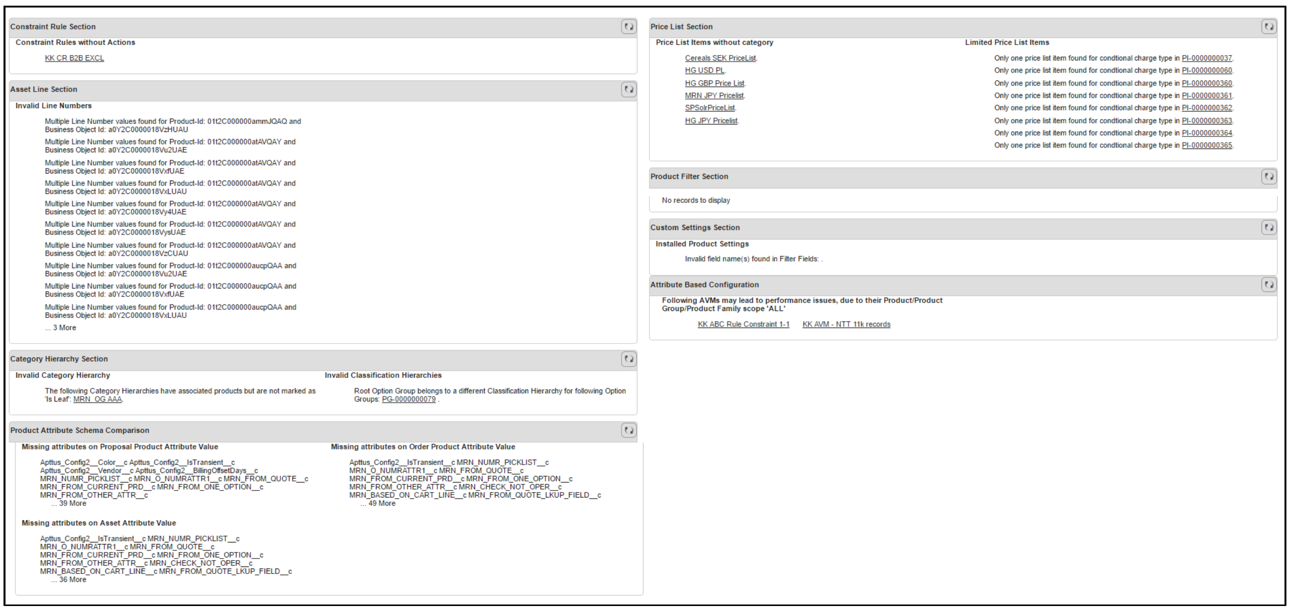
The different promotions applied on your cart and the ones which are pending to be applied on the cart are captured in the form of **Incentive Limit Data**. This also helps in knowing limit details, such as Limit Start Date, Limit End Date, Limit Value and more.

Incentive Groups club all the incentives together under a single row, based on your business requirement, and help you analyse how the incentives applied from a certain Incentive Group are helpful in growing your business.

In case of multiple benefits applied on a single product or cart, the system captures the details of each benefit applied through a promotion in the Adjustment Line Items or Order Adjustment Line Items.

Using Apttus - System Overview

The Apttus - System Overview provides a dashboard to see which Apttus CPQ related configuration is currently incomplete or invalid. As an admin, you can take remedial actions based on the actionable insights provided by the system. System Overview provides a quick snapshot of the overall health of your implementation. You have the ability to see bad data and have a direct link to fix such bad data that resides in the system. Description of such error is also provided for ease of understanding.



Apttus - Systems Overview page lists down the following sections as described in the table:

Section	Sub-Section	Description
Constraint Rule Section	Constraint Rules without Conditions	Lists the constraint rule defined without condition.
	Constraint Rules without Actions	Lists the constraint rule defined without action.

Section	Sub-Section	Description
	Invalid Rule Condition Criteria	Lists the constraint rule that has an action criteria which is not in accordance with type.
	Invalid Rule Action Criteria	Lists the constraint rule that has a condition which is not in accordance with type.
Asset Line Section	Invalid Line Numbers	Lists invalid line numbers. This could be due to the business object being not NULL, and all assets (with the same business object and product) do not have the same line numbers.
	Invalid Bundle Asset Line Item	Lists invalid asset line item due to assets options are present, however, a bundle is NULL.
Category Hierarchy Section	Invalid Category Hierarchy	Lists invalid categories where they do not have a leaf level classification associated.
	Invalid Product Classifications	Lists invalid product classification due to no Product ID associated.
	Invalid Classification Hierarchies	Lists invalid classification hierarchy when option groups are not an immediate child of parent option group or root option group has no parent category associated.
Price List Section	Price List Items without Charge Type	Lists price list item defined without a Charge Type.
	Price List Items without category	Lists price list that has no categories.

Section	Sub-Section	Description
	Invalid Price Dimensions	Lists invalid price dimensions.
	Invalid Product Option Components	Lists invalid product options components without required field, such as no parent product, option group, or component product.
Product Filter Section	Undefined In Search Filter Picklist	
	Deleted Fields	
Custom Settings Section	Config System Properties	Custom view cart fields must be filled in the line item object or view of total custom fields.
	Config Select Product Settings	The selected products column 2/3/4 must be defined as fields in the line item object.
	Config Select Option Settings	Listed options must be fields of product2 object.
	Config LineItem Criteria Fields	Listed options columns must be fields in line item object.
	Config Constraint Criteria Fields	Field name values must be fields of line items objects.

Section	Sub-Section	Description
	Installed Product Settings	Amend change fields must be fields of line items objects.
	Config Custom Display Column	Lists all fields which do not have a display type.
Product Attribute Schema Comparison	Missing attributes on Proposal Attribute Value	Lists all missing attributes for the Proposal Attribute Value.
	Missing attributes on Order Product Attribute Value	Lists all missing attributes for Order Product Attribute Value.
	Missing attributes on Asset Attribute Value	Lists all missing attributes for Asset Attribute Value.
Attribute Based Configuration	-	Lists the AVMs where the scope is defined as ALL either for Product, Product Group, or Product Family. When you have created an AVM with the scope as ALL, the system may cause HEAP errors.

These sections contain direct links that take you to the configuration if there is any such invalid configuration.

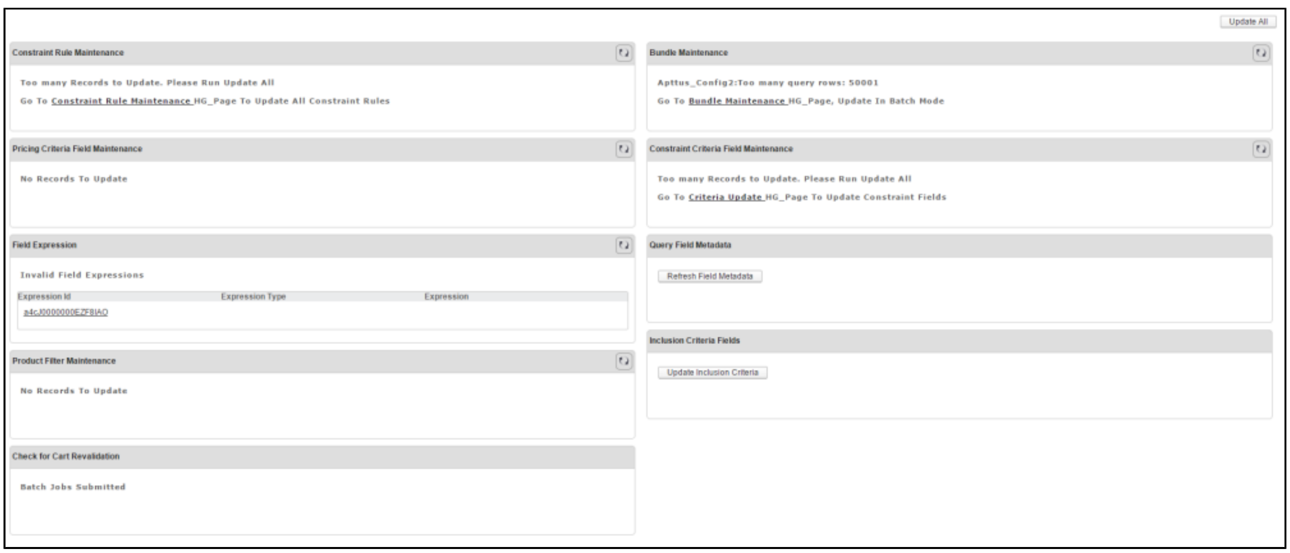
Click  icon to refresh the view of the corresponding section.

Using Maintenance Console

The Maintenance Console provides an overview of maintenance jobs which are currently pending to be ran. In addition, as an admin, you have the ability to refresh the field metadata on the cart pages and update Inclusion Criteria for Search Filter-based criteria, which are based on the Inclusion Rules.

Maintenance Console lists down the following sections for which maintenance jobs are currently pending:

- Constraint Rule Maintenance
- Pricing Criteria Field Maintenance
- Bundle Maintenance
- Constraint Criteria Field Maintenance
- Product Filter Maintenance



You can Update the corresponding maintenance jobs by clicking the relevant link. When there are no pending jobs to be ran, system shows 'No Records to Update' message.

Click **Update All** to run all the maintenance jobs in a fast mode.

⚠

Except category maintenance, all other maintenance jobs have fast mode enabled with a single click update.

Click  icon to to refresh the view of the corresponding section.

Overview of Batch Jobs

Job	Definition
Bundle Maintenance	This job creates a de-normalized view for the Bundle with related option groups and options.
Category Maintenance	This job creates a de-normalized view for the Categories with related products.
Attribute Maintenance	This job creates a de-normalized view for the Attributes with related products associated with the attribute group.
Product Filter Maintenance	This job creates a de-normalized view for the Product filters with related Search attributes, Product fields, and Category Hierarchy fields.
Constraint Rule Maintenance	This job creates a de-normalized view for the constraint rule with related conditions and actions.
Criteria Maintenance • Update Pricing Criteria • Update Constraint Criteria Fields	These jobs optimize the query time by creating indexes for the fields that are referred to in the criteria or expressions.

Job	Definition
• Update Expression Criteria Fields	

Asset Management

Quote-to-Cash is the end to end business process between your customer's interest in a purchase and the revenue that your business realizes from that purchase.

The Quote-to-Cash flow includes creating a Sales Quote, submitting the quote, negotiating and managing the contract, fulfilling the order, tracking payment, and managing your customers' purchases.

In the Quote-to-Cash flow, Asset Management is a set of mission-critical business processes that you can employ to manage your customer's purchased products with a variety of billing models to ensure efficient collections and accounting.

The following table lists and describes industry challenges with Asset Management and the solutions that Apttus provides for each challenge.

Challenges with Asset Management	Apttus-provided solutions
Asset Management can be a complex coordination of activities including custom design and configuration. Depending on your business requirements it could be, • scheduling milestone-based manufacturing processes • software provisioning • equipment installation • scheduling services teams. If customers update their orders, fulfillment becomes even harder to manage.	Asset Management is based on the order information that you have already defined in your sales quote and resulting contracts. Asset Management then, becomes very easy when you integrate Apttus CPQ with Apttus Contract Lifecycle Management (CLM). Historically, Assets have been managed using an Enterprise Resource Planning (ERP) application. Because it is important for you to capture on-the-fly changes to orders and integrate your Order Management processes with customer facing applications like Contract Management solutions, assets or purchased products are now also managed within a CRM system, or in multi-functional CPQ applications.

Challenges with Asset Management	Apttus-provided solutions
If a customer has multiple orders or changes, invoices must be coordinated so your customers do not receive multiple invoices.	Renewals, changes, swaps, terminations, and additions to an asset increase the complexity of invoices. You can use Asset Management to generate an accurate, up to date, and easy to understand invoice so your customers understand it easily, pay quickly, and have a positive experience.
Your finance team must analyze transaction and revenue information to device the best revenue schedules that determine when revenue is recognized. Revenue recognition depends on when your customers receive their products, services and subscriptions.	Revenue recognition can be a challenge, but it’s much easier when Finance can automatically generate the appropriate revenue schedules for products, services, and subscriptions based on contracts. With Apttus CPQ, you can verify if services are rendered or products delivered, and then see how those impact your customers' billing schedules and the overall revenue forecast.

You can define the Asset Management functions with different data objects to track Quote and Contract details until an order is fulfilled. With Asset Management you can

- make it easy for customers to add more products and services
- align all product changes to your customer’s bill cycle
- drive charge pro-rations and pro-ration credits
- customize the renewal process that best suits your organizational needs

The following table lists the benefits Apttus Asset Management has over ERPs.

Challenges with ERPs	Solutions by Asset Management
Do not understand recurring revenues or distinguish between monthly recurring revenues and annual recurring revenues.	Works with accounting solutions and meet the financial requirements unique to both one-time purchases and subscription businesses.

Challenges with ERPs	Solutions by Asset Management
Do not handle pricing and packaging changes like upgrades, downgrades or increments.	Can quickly modify your pricing and packaging options irrespective of the number you increment or decrement.
Do not tell you how many active customers you have, or your rate of customer retention.	The reporting feature can detail your renewal rates and help you up-sell, cross-sell, and formulate strategies to increase customer retention.
Do not give you a unified view of how your business is doing.	Works seamlessly with Apttus CPQ, CLM, and Billing Management to bring awareness of information including renewal plans and usage statistics to every department in your organization.

Whether it is a one-time order, a subscription, or a usage-based service you sell to your customers, you can use Asset Management to manage any purchased product or service for your customers.

Asset-Based Ordering functionality is now built into Apttus CPQ . With the Asset-Based Ordering functionality, you can service and manage existing orders based on your customers' requirements.

With ABO, you can perform transactions for Purchased Products using cloud-enabled systems that operate on a massive scale and real-time speed. With Asset-Based Ordering, you can perform complex functions such as suspensions, mid-cycle upgrades, downgrades, add-ons, and cancellations, all on a single, user-friendly interface.

With Asset-Based Ordering you can

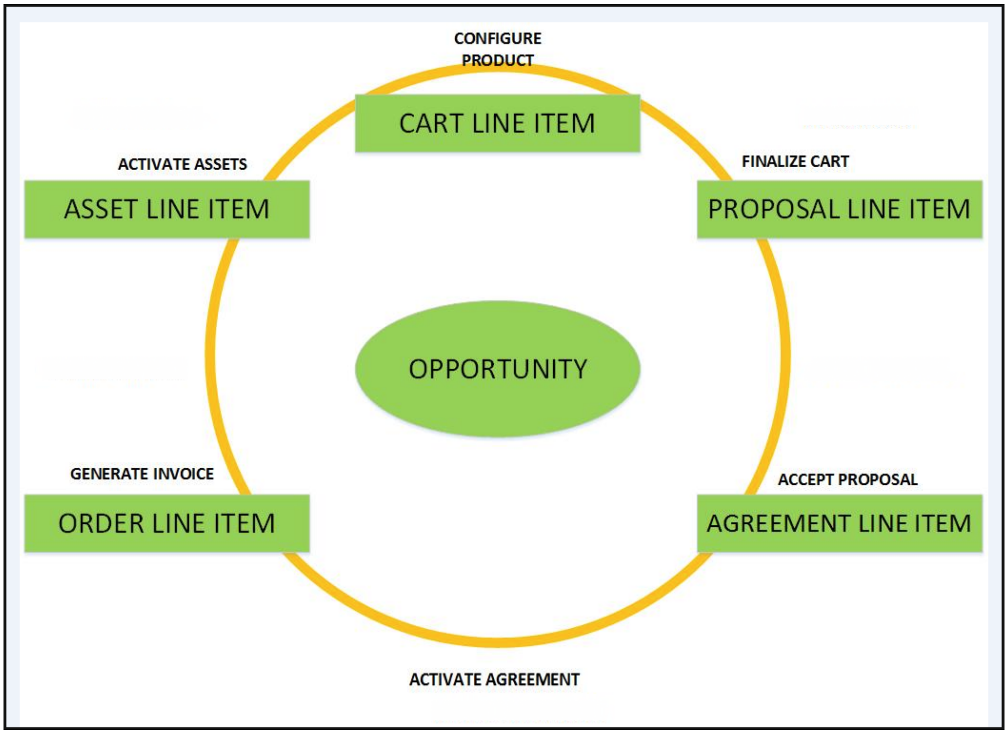
- work seamlessly with CRMs to bring subscription awareness to your front office
- unify your commerce, billing, and finance processes so customer relationships take center stage
- increase recurring revenue that you earn from existing customers by retaining them with renewals, incentives, rebates, and promotions

Asset-based Ordering adds to the robust CPQ functionality and the larger Quote-to-Cash experience by giving you a quick return on your investment and quickly increasing both, revenue and customer base.

Assets and Asset Line Items

Any product or service that your customers buy, become that customer's assets. While your organization may have a unique and customized Order Fulfillment process, all the processes and procedures in Apttus Asset Management start with the Line Item in an order.

The following image describes the life-cycle of a Line Item in the Quote-to-Cash flow.



After your customers agree to purchase a product, the product changes from a Proposal Line Item to an Order Line Item. The status of the Quote/Proposal in this stage is **Accepted** and the order is now ready to be invoiced.

After you receive payments for an order is when the order line item changes to an **Asset** line item.

As an Administrator, you must configure Apttus CPQ to enable Customer Service and Sales Representatives in your organization to be able to perform the following actions to manage assets .

1. Generate asset records from orders
2. Renew the term for an asset
3. Change the **Term**, **Quantity**, **Options**, and **Attributes** of a product
4. Cancel products and subscriptions
5. Swap one purchased product with another

6. Create quotes for new products and services based on assets
7. Create quotes to modify existing products and services
8. Get visibility into the asset life cycle during customer interactions

Before you start to define the mechanism that Customer Service and Sales Representatives must use to manage a customer's assets, you must understand how they use Asset-Based Ordering.

Asset Management Flow

Before you start configuring Asset-Based Ordering, you must understand how the Sales and Customer Service Representatives in your organization will use the ABO functionality.

There are total 3 flows that you can consider before starting the configurations for using ABO. Ensure that you understand and choose one of the flows to implement error-free ABO.

1. Quote/Proposal Flow**:** This is one of the signature events in Asset-based Ordering involving Quote/Proposal.This is the normal flow of going through the Quote/Proposal to an Order and then to the Assets. Once you accept a Quote/Proposal, the Order is generated. Upon activation of the order, assets are created.
2. CSR Flow**:** This is the flow designed to enable the Sales Rep to skip the creation of a Quote /Proposal. No signature event, such as creation of an Agreement or a Quote/Proposal is involved. A new button called **Asset Manager** is configured on the Accounts page to help the Sales Rep navigate directly to the Installed Products page.
3. Contract Flow**:** This is one of the signature events in Asset-based Ordering involving Contracts/Agreements. The assets are created through Contracts/Agreements using the normal Contract lifecycle flow. Once you finalize a Contract (containing Agreement Line Items), the corresponding Asset Line Items are created.

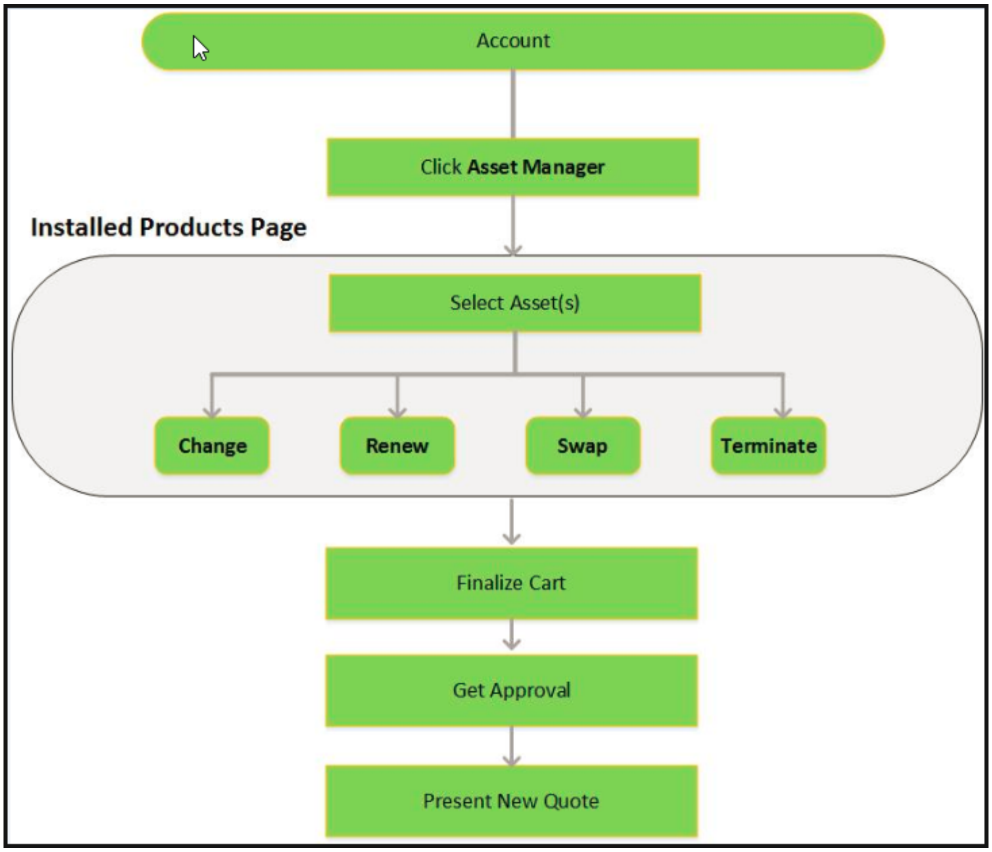
Quote/Proposal Flow

In this flow, the Sales Rep configures the Quote/Proposal and finalizes a configuration. Once the configuration is finalized, the Sales Rep presents the configuration to the customer. Upon acceptance from the customer, the Sales Rep clicks the **Accept** button on the Quote/Proposal. An order is created for this Quote/Proposal, which upon activation, creates the Asset Line Items with the *New* status. The Sales Rep can configure the Quote/Proposal again and navigate to the Installed Products page to view all the active assets. No custom triggers are required here.

CSR Flow

There can be scenarios when your Sales Rep does not want to configure the Quote/Proposal again to land on the Installed Products page. For these type of scenarios, you can configure the **Asset Manager** button on your Account page. Clicking **Asset Manager** creates an Order and navigates you directly to the Installed Products page. In this flow, you skip the navigation of going to the Installed Products page through the **Quote/Proposal > Configure Products**. You just have to configure a custom formula button to achieve this, explained here.

The following image illustrates the CSR flow from the Sales and Customer Service perspective.



Contract Flow

A contract contains Agreement Line Items, Asset Line Items or Order Line Items. In this flow, you follow the normal Contract Lifecyle flow of creating a Contract, presenting it to your customer and finalizing the Contract. Here, the Asset Line Items are created when the Sales Rep finalizes and activates a Contract. Apttus CPQ now provides the ability to synchronize the changes in Assets with the changes in a Contract. The changes you make inside a Contract (such as change in Contract End Date) will be reflected in the Assets contained in that Contract. Similarly, when you perform actions on your purchased Assets (such as changing the Quantity of an Asset), these changes will be reflected in the Contract.

See Configuring the Custom Settings for Asset Management Flow to learn how to configure the above flows.

Packages required for implementing Asset-Based Ordering

For implementing Asset-based Ordering, follow the instructions of installing a package from the Install Center explained here. Ensure that you have installed the following packages in your environment. Always check the latest Release Notes to get the latest package versions.

For the Quote/Proposal Flow,

Order	Package	Install Center tab to access the package
1	Apttus Proposal Management	CPQ
2	Apttus Configuration & Pricing	CPQ
3	Apttus CPQ API	CPQ
4	Apttus Quote/Proposal-Configuration Integration Integrations	Integrations
5	Apttus Quote/Proposal-Asset Integration	Integrations

For the Contracts Flow,

Order	Package	Install Center tab to access the package
1	Apttus Contract Management	Contract Management
2	Apttus Quote/Proposal-Contract Integration	Integrations
3	Apttus Contract-Configuration Integration	Integrations

Configuring Asset-Based Ordering

As an administrator, you must enable Customer Service and Sales Representatives in your organization to **Change**, **Swap**, **Renew**, and **Terminate** a customer's assets. To this end, you must complete the following procedures in the listed order.

Mandatory Configuration

The following steps are the mandatory configuration steps which are required to implement ABO successfully.

1. Granting appropriate access and permissions to Managers, partners, other administrators, Customer Service, and Sales Representatives
2. Creating the flow settings
3. Configuring the Installed Products Settings
4. Configuring the Custom Settings for Asset Management Flow
5. Customizing Display Actions for the Installed Products page
6. Defining the look and feel of the Installed Products page
7. Setting up Asset-based Pricing
8. Configuring Lookup Field Settings

Optional Configuration

The following configuration steps are optional. You may implement these steps as per your business requirements.

1. Defining the Actions you want to allow for each Asset Line Item
2. Integrating Assets with Contracts
3. Creating an Approval Workflow for **Renew**, **Change**, **Swap**, and **Terminate**
4. Creating a Replacement Rule to enable the Swap feature
5. Configuring the Custom Attributes Page

The values you enter for the fields in the pages listed above will determine how Customer Service and Sales Reps in your organization manage a customer's assets.

Configuring the Custom Settings for Asset Management Flow

For the different Asset Management Flows explained here, you must configure different custom settings.

To enable the Quote/Proposal Flow

1. Go to **Setup > Build > Customize > Develop > Custom Settings > Proposal System Properties > Manage**.
2. Click **Edit** next to **System Properties**.
3. Search for **Auto Create Order** check box and select it. Refer Proposal System Properties for details.
4. Click **Save**.

To enable the CSR (Order) Flow

Asset Manager is a simple formula field that enables the creation of an order header which in turn is used as a container to manage a customer's assets without a proposal.

1. Click **Setup > Create > Objects**.
2. Select the **Account** object and go to the **Custom Fields & Relationships** related list.
3. Create a new formula field with the label **Asset Manager** and *Text* as **Return Type**.
4. In the advanced formula editor, enter the query string with the following parameters.

```
HYPERLINK("/apex/Apttus_Config2__AccountOrderCreate?id=" & Id &
"&method=csrFlow&flow=csrFlow&priceListId=<pricelistid>&launchSt
ate=assets&autoFinalize=true&activateOrder=true&retId=" & Id ,
IMAGE("", "Asset Manager"),"_self")
```

In the above formula, replace **<pricelistid>** with the *ID* of your desired Price List. For the flow parameter, specify the name of the Flow that you have defined in the Flow Settings. To support this button across different Price Lists, you must create separate buttons for each Price List.

5. Click **Save**.

To enable the Contract Flow

1. Go to **Setup > Build > Customize > Develop > Custom Settings > Comply System Properties > Manage**.
2. Click **Edit** next to **System Properties**.
3. Search for **Auto Create Order** check box and select it. Refer Comply System Properties for details.
4. Click **Save.**

> ⚠ Do not select **Auto Activate Order** check box of **Comply System Properties** and **Proposal System Properties** simultaneously. This might give you unexpected results.

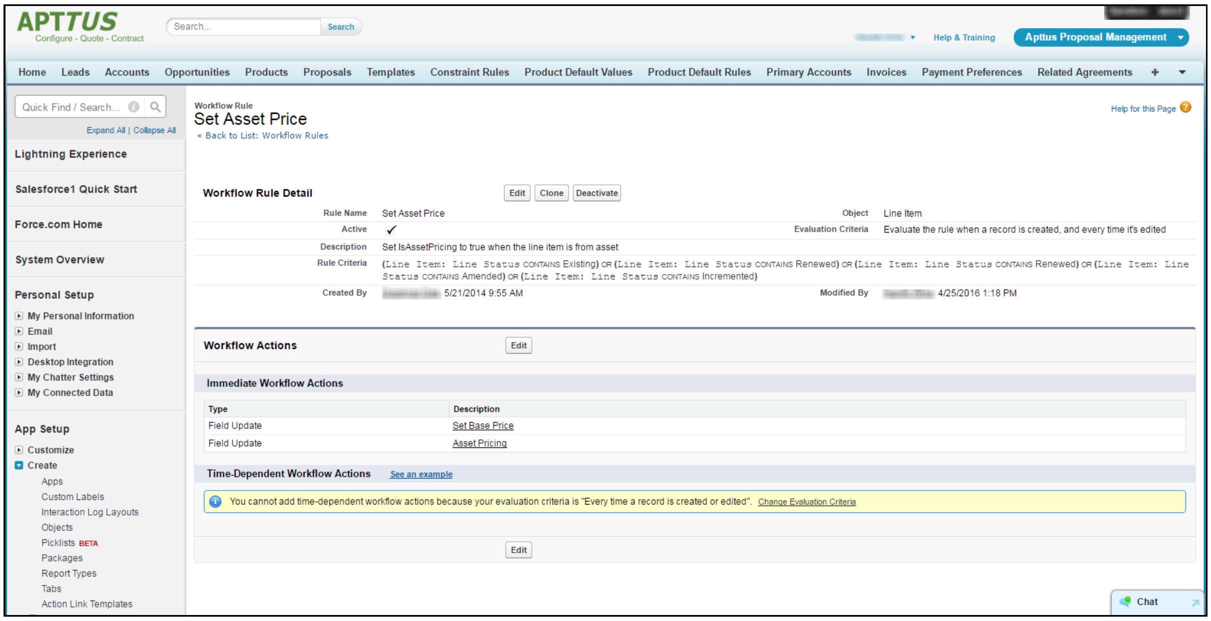
For detailed information on the behavior of these two check boxes, see Integratng Assets with Contracts.

Setting up Asset-based Pricing

The price loaded in an asset is based on the price mentioned in its Price List Item. As an administrator, you can decide if you want to retain the asset price or set the new base price in the further transactions on the Installed Products page. A workflow is used to set the the above actions for the asset pricing.If the **IsAssetPricing** field of the Line Item object is set to *True*, the system retains the price of the Asset Line Items for any future renewals or amendments.

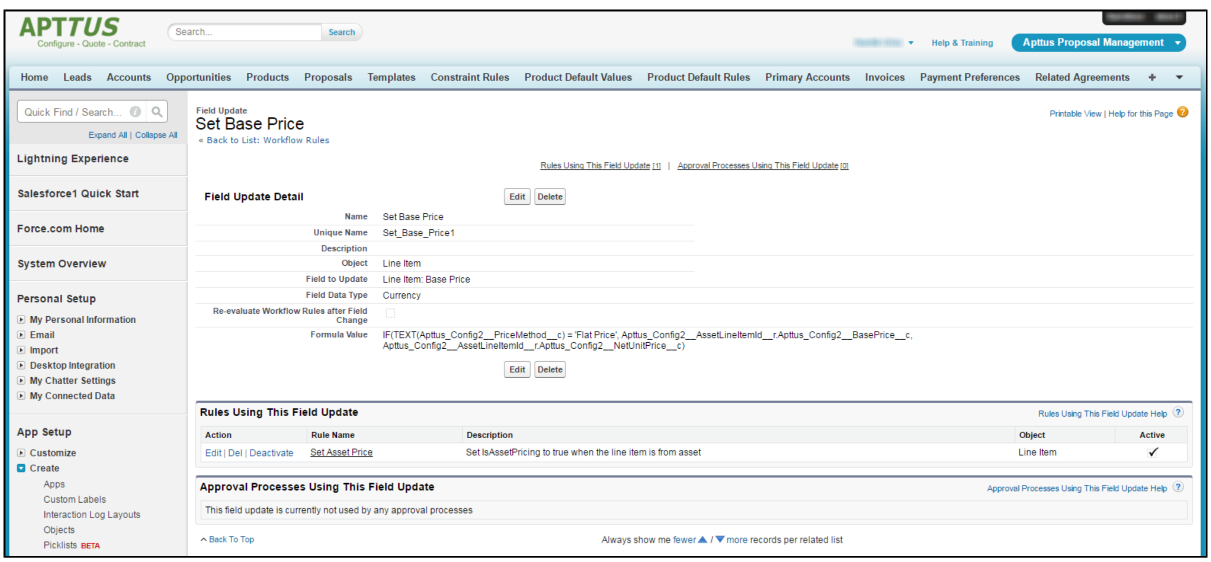
For example, the pricing for a cell phone plan has increased from $20 to $30 per month, and the Price List Item record is updated to reflect the new list price. However, the selling company has decided that customers who already have the product installed as an asset will not be charged the new price for renewals, amendments, etc. These customers shall be allowed to purchase the same subscription at the base price at which they purchased it originally.

You must create a Workflow Rule for the Line Item object, as described in the Salesforce Documentation.



For the Workflow actions,

1. Set the IsAssetPricing check box to *True*.
2. Set the formula for Base Price (depending on your requirement).



Ensure that all the Options products of your Bundle follow the Asset Pricing if you have configured a Workflow Rule.

Configuring the Installed Products page

The Sales and Customer Service Reps use one of the flows to navigate to the Installed Products page. You must complete the following procedures to define the content and aesthetics of the Installed Products.

1. Customizing Column Settings for the Installed Products page.
2. Defining the number of products displayed on the Installed Products page.
3. Asset Visibility on the Installed Products page.

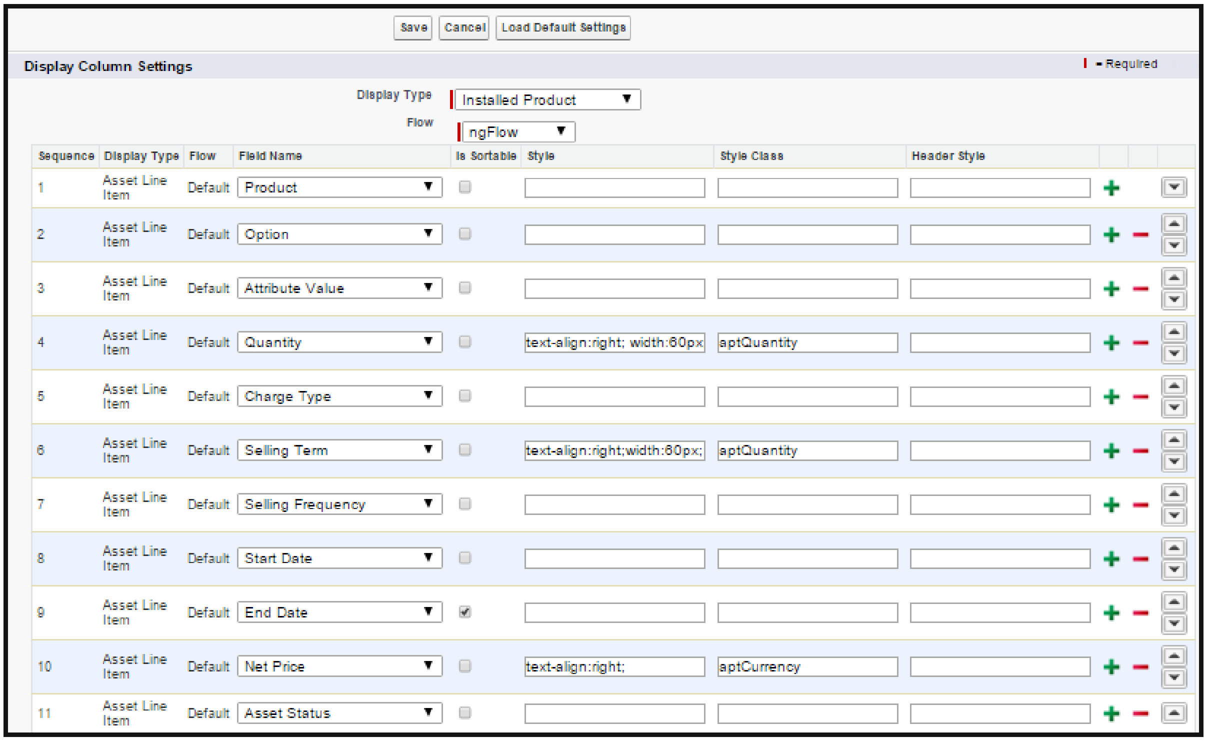
The values you define for the pages listed above, determines how the Sales and Customer Service Reps access and use the Installed Products page.

Customizing Column Settings for the Installed Products page

You can customize the Installed Products page to suit both, your business requirements and the ease-of-use for the Sales and Customer Service Representatives in your organization.

To configure these settings,

1. Click **All Tabs** (**+**) > **Config Settings** > **Display Column Settings**.



2. Add the **Product** field first, followed by the **Option**, **Attribute**, and **Charge Type**.

> ### Note
>
> You must also ensure that the Product field in the Cart page settings in NOT EDITABLE and always the first column.

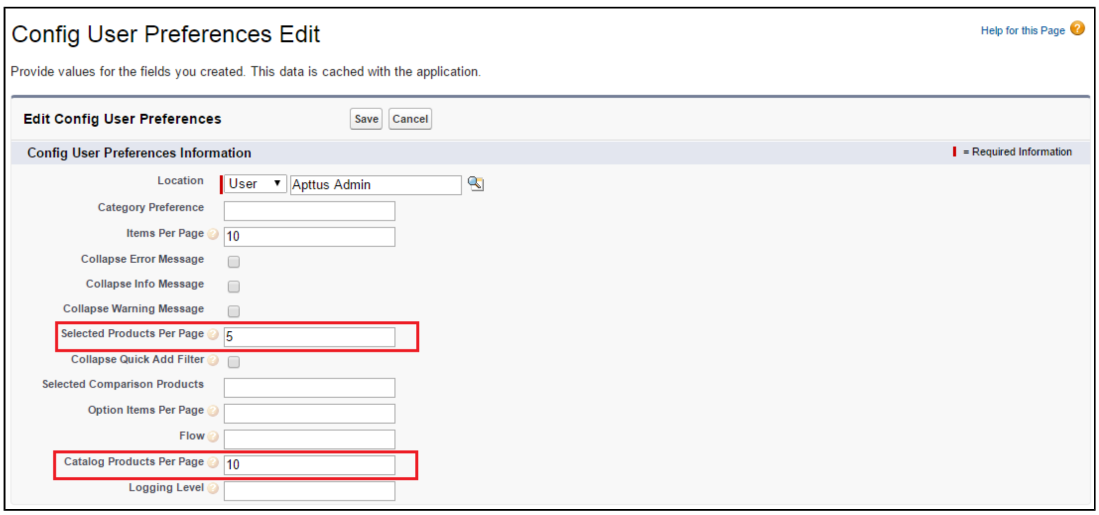
3. Click **+** to add any subsequent fields that you want on the Installed Products page.
4. Deselect the **Is Sortable?** check box for all the fields you have added.
5. Click **Save**.

Defining the number of products displayed on the Installed Products page

For the New UI, you can use the following custom setting to customize the number of products on the Installed Products page. This number can be for a specific user, as per the user preference. The products are divided across the pages using the Pagination concept.

To enable the above enhancement:

1. Go to **Setup > Develop > Custom Settings** and search for **Config User Preferences.**
2. Click **Manage** next to **Config User Preferences**.
3. For the required user, do any of the following:
   Click **View** to view the details and click **Edit.**
   - Or -
   Click **Edit** to edit and make changes to the existing user preferences.



4. In the **Catalog Products Per Page** field, enter the number of products that you want to display on the Catalog page per page. Valid values are *10*, *20*, *50*, and *100*. This setting is available specially for the New UI.
   In the **Selected Products Per Page** field, enter the number of products that you want to display in the mini cart on the Catalog page. Valid values here are *5*, *10*, *15*, *20*, and *25*. This setting is available for both, the enterprise CPQ and the New UI in CPQ.
5. Click **Save**.

Now, when you browse the catalog page as a specific user, you will find the specified number of products on a single page and the rest of the products are divided in multiple pages using Pagination.

Asset Visibility on the Installed Products page

You can navigate to the Installed Products from the Catalog page, Cart page, or through the Account page with the help of different formula buttons. You can control the assets (bundles, options and attributes) dispalyed on the Installed Products page. You can also see critical attribute values at both, the bundle and sub-bundle level when expanding a Bundle product on this page.

Filtering the assets on the Installed Products page

- **Search Filters:** You must create Search Filters for the Installed Products page to enable the Sales or Customer Service Reps in your organization to search for specific purchased products that customers ask to be changed, swapped, or terminated.
- Callback class**:** You must use the code in the sample AssetLineItemCallback class to further filter the assets on the Installed Products page.

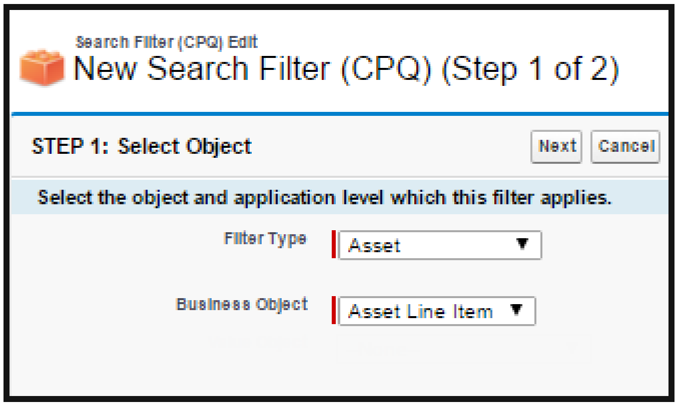
To customize the Filter Fields on the Installed Products page

1. Click **All Tabs** ( ) **> Config Settings > Installed Products Settings**.
2. In the Filter Fields, specify the API names of the Asset Line Item fields that you want to use as filter fields.
   Currently, the fields of type *Text*, *Picklist, Multi-picklist, Date, Datetime* and *Boolean* are supported.
3. Click **Save**.

These fields will show up as filter fields on the Installed Products page. These fields work as temporary filters and you can change the values in these filters dynamically on the Installed Products page.

To create a Search Filter on the Installed Products page

1. Click **All Tabs** ( ) > **Search Filter (CPQ)** > **New** .



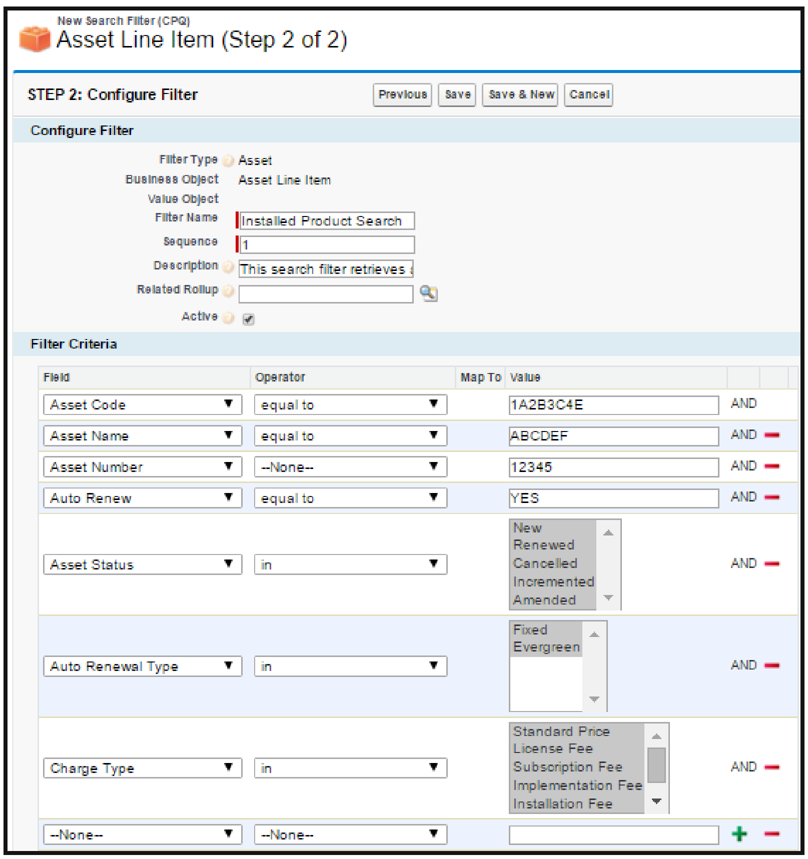
2. Enter values for the following fields.
   Filter Type - Asset
   Business Object: Asset Line Item
3. Click **Next.**
4. In the Configure Filter section, enter values for the fields described in the following table .

Field	Description
Filter Name	Enter a unique name for this search filter.
Sequence	Enter the sequence in which the Search filter is displayed on the Installed Products page.
Description	Describe the function and purpose of this search filter to avoid duplication.
Related Roll up	Lookup and select a pre-defined roll up on which you want to base your search results.
Active	Select the check box so the Sales and Customer service representatives can use the filter.

5. In the Filter Criteria section, click **+** to add the fields you want to include to your search criteria . Use the available operators to establish the relationship with the value you define for each field.
   For example, you want the Sales and Customer service representatives in your organization to be able to search for assets based on the following parameters.

-
    - Asset Code
    - Asset Name
    - Asset Number
    - Asset by Status
    - Assets with **Auto Renew** enabled
    - Assets of any Auto Renew Type

You can add fields to the Search Filter and define values for these fields that are based on your business requirement.



6. Click **Save**.

When you specify the value in a filter field, only those records that match the criteria of the Search Filter are displayed. For example, for **Start Date** and **End Date** filter fields, if you specify the **Start Date** as *07/01/16* and **End Date** as *07/31/16*, the system displays the assets having **Start Date** as 07/01/16 or **End Date** as 07/31/16 and not the assets falling in the range of these two dates.

Using Asset Filter Callback class

Callback Class for Asset Visibility

You can use the code in the custom callback class to set the criteria based on which you want to filter the assets on the Installed Products page. You have to enter the name of the custom callback class that you use in the Custom Classes.

To add the custom class

1. Go to **Setup > App Setup > Develop > Apex Classes**.
2. Click **New**.
3. Enter the sample callback class code.

   This is just a sample callback custom class to control which assets are displayed on the Installed Products page. You may change the code as per your requirement.

```
global with sharing class AssetLineItemCallback {
    private String assetSearchFilter = null;
    private List<String> assetSearchScope = null;
    private ID priceListId = null;

    /**
     * Class Constructor specifying initial values
     */
    public AssetLineItemCallback() {
    }
```

```
    /**
     * Callback at the beginning of the asset selection call.
     * Use the start method to initialize state
     * @param cart the cart object or null if there is no cart
context
     * @param assetSearchFilter the preset static filter used
in the asset search or null if there is no preset filter
     * @param assetSearchScope the list of asset fields to
match the search text or empty to use the default list of fields
     */
    global void start(ProductConfiguration cart, String
assetSearchFilter, List<String> assetSearchScope) {
        this.assetSearchFilter = assetSearchFilter;
        this.assetSearchScope = assetSearchScope;
        this.priceListId = cart.getConfigSO().
PriceListId__c;
    }

    /**
     * Callback to return part of SOQL filter clause
     * This filter is used in listing installed products
     * @param accountId is the context account id
     * @return The query filter is like the following.
     *         Name LIKE 'A%' AND Quantity__c > 100
     *         Id IN ('000123', '000124')
     */
    global String getQueryFilter(ID accountId) {
        return '';
    }

    /**
     * Callback to return the filter expression for the asset
query where clause
     * This filter is used in listing installed products
     * @param params the parameters for the method
     * @return the filter expression or null to use the default
filter.
     * e.g. Name LIKE 'A%' AND Quantity__c > 100
     *         Id IN ('000123', '000124')
     */
    global String getFilterExpr(CustomClass.ActionParams
params) {
        System.debug('filter params = ' + params);
        String clause = '';
        if (params.AccountId != null) {
            String ns = RuntimeContext.getNSPrefix();
```



```
            clause = ns + AssetLineItem.FIELD_NAME_ACCOUNT_ID +
' = \'' + params.AccountId + '\' ';
            if (priceListId != null) {
                clause += ' AND ' + ns + AssetLineItem.
FIELD_NAME_PRICELIST_ID + ' = \'' + priceListId + '\' ';
            }
        }

        if (systemUtil.nullOrEmpty(clause)) {
            if (!SystemUtil.nullOrEmpty(assetSearchFilter)) {
                //append static asset filter
                clause += ' AND '  + assetSearchFilter;
            }
            System.debug('AssetFilter = ' + clause);
            return '(' + clause + ')' ;
        } else {
            System.debug('AssetFilter is empty');
            return '' ;
        }
    }

    /**
     * Gets the asset search scope
     * @return the asset search scope or null to use the
default asset search scope
     */
    global List<String> getAssetSearchScope(){
        return this.assetSearchScope;
    }

    /**
     * Callback after the filter is used
     * Use the finish method to release state
     */
    global void finish() {
    }

}
```

4. Click **Save**.

> The Class Name will be auto-populated. If you are using the Apttus Custom Class code then ensure the name is *AssetLineItemCallback*.

To add the name of custom callback class, go to **Setup > App Setup > Develop > Custom Settings** and click **Manage** besides **Config Custom Classes**. Click **Edit** for **Custom Classes** and enter *AssetLineItemCallback* or the name of your custom callback class in Asset Line Item Callback Class.

Configuring Lookup Field Settings

Each new lookup field shown in the AngularJS cart needs a corresponding lookup field settings record. You must follow the explanation described here to create a new record for a lookup field settings.

Billing Preference look up setting.

Field	Value
Name	Billing Preference
Display Columns	Name
Filter Criteria	Optional, to further filter the records
Lookup Field Name	API name of the Billing Preference field, *Apttus_Config2__BillingPreferenceId__c*.
Object Name	API name of the Line Item object, *Apttus_Config2__LineItem__c*

Agreement ID look up settings

Field	Value
Name	Agreement ID
Display Columns	Name
Filter Criteria	Optional, to further filter the records
Lookup Field Name	API name of the Agreement field, *Apttus_CMConfig__AgreementId__c*.
Object Name	API name of the Line Item object, *Apttus_Config2__AssetLineItem__c*.

Accessing the Installed Products Page

To access the Installed Products Page, you must create a new quote. When you configure products, the system creates a new cart with an Account ID attached to it.

Add a few products to the new quote and finalize the cart and accept the quote. Make sure that the quote status has changed to *Accepted*. After you click the **Installed Products** button, the Installed Products page displays all the asset line item records from the account information linked to the current cart.

To configure direct navigation to the Installed Products page from the Quote /Proposal page

1. Click **Setup** > **Create** > **Objects**.
2. Select the Quote/Proposal object and from the Custom Fields & Relationships related list, select **Configure Products** with the Installed Package of Apttus Quote/Proposal-Asset Integration.
3. Click **Edit** and replace the query string with the following parameters.

   For the classic CPQ,


```
IF ( LEN( Apttus_QPConfig__PriceListId__c ) > 0 , HYPERLINK("
/apex/Apttus_QPAsset__ProposalConfiguration?id=" &Id, IMAGE("
/resource/Apttus_QPConfig__Button_Configure", "Configure
Products"),"_self"), NULL)
```

For the new UI,

```
IF ( LEN( Apttus_QPConfig__PriceListId__c) > 0 , HYPERLINK("
/apex/Apttus_QPConfig__ProposalConfiguration?id="&Id+"&flow=ngfl
ow&launchState=assets",
IMAGE("/resource/Button_Configure", "Configure Products"),"_self
"), NULL)
```

4. Click **Save**.

On the Quote/Proposal page when you click the **Configure Products** button, the Installed Products page appears instead of the Catalog page.

To access the Installed Products page directly from the Account page

1. Go to **Setup** > **Create** > **Objects**.
2. Select the **Account** object and from the **Custom Fields & Relationships** related list, create a new custom formula field.
3. In the Advance Formula editor, enter the query string with the following parameters.

```
HYPERLINK("/apex/Apttus_Config2__AccountOrderCreate?id=" & Id &
"&method=csrFlow&priceListId=<pricelistid>&flow=csrFlow&launchSt
ate=assets& activateOrder=true&retId=" & Id , IMAGE("", "Asset
Manager"),"_self")
```

   In the above formula, replace **<pricelistid>** with the *ID* of your desired Price List. For the flow parameter, specify the name of the Flow that you have defined in the Flow Settings.

4. Click **Save**.
   This button creates an order specific to a Price List and Account ID. You may have to create different asset manager buttons for different accounts.

Defining Permitted Actions for Assets

To meet specific business requirements, you can select which of the **Renew**, **Change**, **Swap**, **Terminate** actions are available for a sales user to perform for each Asset Line Item.

For example, you are an administrator for a news publication. Your company has several digital offerings that subscribers buy on a monthly term. One subscription has reached its end-of-life. To prevent you sales representatives from renewing a customer's subscription for this product, you can exclude **Renew** from the list of Allowed Actions.

To define the Allowable Actions manually,

1. Select the Asset Line Item and click **Edit**
2. From the **Allowed Actions** field, select the actions you want to enable and move them to the Chosen column.
3. Click **Save**.

Note

If you exclude any action, the corresponding button is grayed out on the Installed Products page. If you do not select any action, by default, all the action buttons are available to the sales user for use.

For **Renew**, **Terminate** and **Change** actions, you can select multiple Asset Line Items. For **Swap**, you can select one Asset Line Item at a time.

Integrating Assets with Contracts

Apttus CPQ provides the ability to synchronize the changes in Assets with the changes in a Contract. The changes you make inside a Contract (such as change in Contract End Date) are reflected in the Assets contained in that Contract. Similarly, when you perform actions on your purchased Assets (such as changing the Quantity of an Asset), these changes will be reflected in the Contract.

For Contracts, the actions include: **Renew** and **Cancel**. For Assets, the actions include: **Change**, **Swap**, **Terminate**, and **Renew**.

There are basically 4 process flows you must consider:

1. **Contract Flow:** In this case, only Contract and no Quote/Proposal is involved. A contract contains Agreement Line Items, Asset Line Items or Order Line Items. These Line Items are created when a contract is finalized.
2. **Contract + Quote/Proposal Flow**: In this case, both, Contract and Quote/Proposal are involved. A contract contains Agreement Line Items, Asset Line Items or Order Line Items. These Line Items are created from the Proposal Line Items when a Quote is accepted.
3. **Normal Flow:** A Quote/Proposal document contains Asset Line Items or Order Line Items and these are created when a Quote/Proposal is accepted.
4. **CSR Flow:** In this case, only an Order is involved. An Order is created when the Sales Rep clicks **Asset Maneger** button on the Account.

You must set the custom setting **Auto Create Order** under **Proposal System Properties** and **Comply System Properties** to implement the different flows as described below:

Flow Number	Auto Create Order (Proposal System Properties)	Auto Create Order (Comply System Properties)	System behaviour
Contract Flow	False	True	Contract is created without a Quote/Proposal. The Asset and Order Line Items are created when you activate a Contract.
	False	True	

Flow Number	**Auto Create Order (Proposal System Properties)**	**Auto Create Order (Comply System Properties)**	**System behaviour**
Contract + Quote /Proposal Flow			A Quote/Proposal is created followed by a Contract. The Asset and Order Line Items are created when you activate a Contract.
Normal Flow and CSR Flow	True	False	The acceptance of a Quote/Proposal creates the Asset and Order Line Items. Order Line Items and Asset Line Items are created by clicking Asset Manager on the Account.
-	False	False	No Asset and Order Line Items are created.
-	True	True	This is not a valid combination unless there are independent flows which is unlikely.

Ensure that you have installed Apttus Contract Management package, version 8.4.0325 (8.325.1) to use the above feature.

Creating an Approval Workflow for Asset-Based Ordering

You can enable Apttus Advanced Approvals for your purchased products, using a few configuration steps.

Following are the sample use cases when you would want to trigger approvals:

- When your customer requests for the Renew of an asset.
- When your customer requests a change in the fields of an asset, such as Quantity or Discount Type, Discount Amount, and more.

Refer to Apttus Advanced Approvals Guide to understand and implement Advanced Approvals for your object, Assets Line Items. Since this is a custom object, you must ensure to use *Apttus_Config2__AssetLineItem__c* as the API name for customization in all of the procedures mentioned here.

Once enabled, you see **Submit for Approval** button on the Cart, every time you perform action on the asset.

Creating a Replacement Rule to Swap Assets

In order to use the **Swap** action for your purchased products, you must create a Constraint Rule of **Action Type** as *Replacement.* This rule fetches the available product in exchange of the product that you want to swap. Specify the product which you want to swap for in the Constraint Rule Condition and the product with which you want to swap to in the Constraint Rule Action.

See Constraint Rule sections to configure a Replacement Rule.

Configuring the Custom Attributes Page

To enable Sales and Customer Service representatives to select options for Installed Products, view the pre-defined Custom Attribute page for while navigating from the Installed Products page, you must use the *Apttus_Config2__ProductAttributeDetail3* Visual force page to create a Custom Attribute page .

To customize your flow settings, use the following procedure.

1. Go to **All Tabs ( ) > Config Settings** and click **Flow Settings**.
2. In the **Product Attribute Detail Page** field, enter the name of your custom attribute page.
3. Click **Save**.

Setting up Opportunity-based Renewals

Concept

Speaking of customer retention for subscription businesses, a huge margin of revenue comes from the renewals. According to industry experts, retaining existing customers is more cost-effective than acquiring new customers. This means Renewals are a vital source of recurring revenue. However, often the renewals get delayed or lost in the process of gaining potential customers.

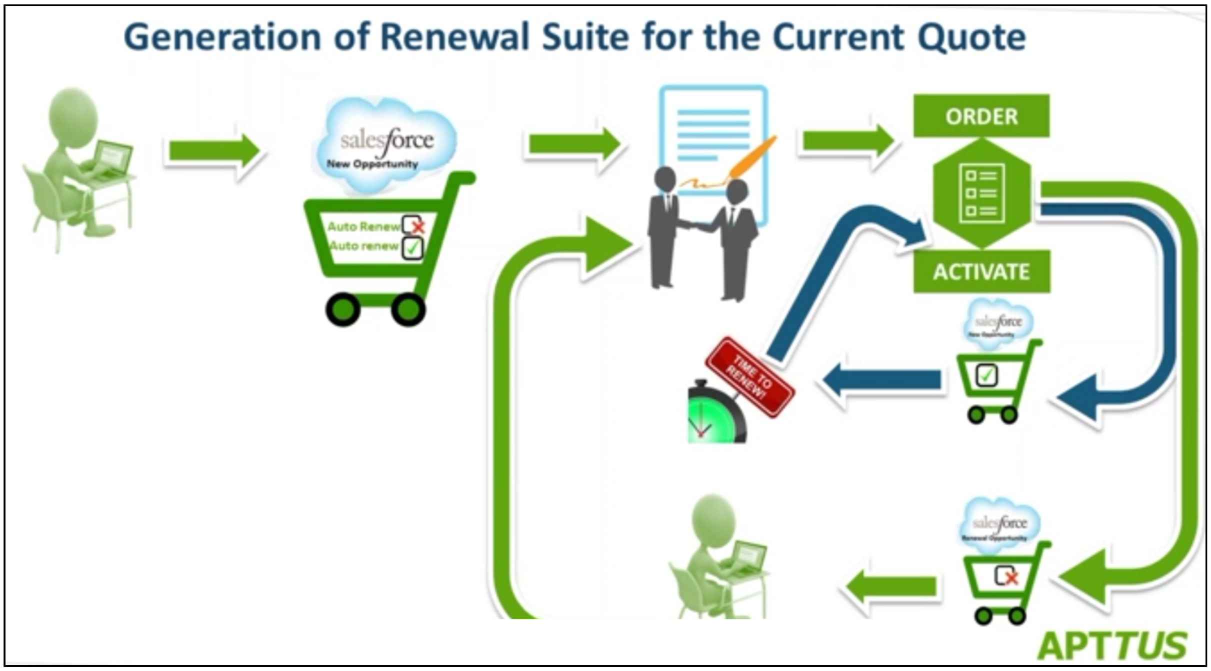
When you create an Opportunity, you ensure that you have a mechanism to track your sales or deals. Opportunity-based renewals help you auto generate renewal opportunities based on the Order fulfillment or expiration of the subscription. Through Opportunities, you can forecast and analyze the Sales pipeline. With Opportunity-based renewals, you get the finer visibility into the pricing and configuration of assets within the renewal pipeline. You benefit with,

- Accurate forecasting on the asset state and sales pipeline
- Faster and simpler processing with automated Renewal
- Automatic closure of Opportunity on asset expiry

Overview

The diagram shows the creation of a renewal opportunity on the closure of the deal.

1. Sales Rep creates an Opportunity by creating a Quote and adding products to the cart.
2. Some of the products may be eligible for auto renewal and some may not. This is differentiated from the **Auto Renew** flag on the product PLI.
3. After finalizing the cart and accepting the Quote, the Order and assets are generated.
4. System triggers Renewal Opportunity creation on Order activation.
5. Sales Rep will see that two renewed opportunities are created,
    a. Opportunity for auto renewed lines
    b. Opportunity for non-auto renewed lines
    This also depends on the Grouping you choose to apply for creating renewal opportunities. Check **Renewal Group Fields** here.
6. Sales Rep can open the Quote associated with the renewed opportunity and verify the asset configuration on the cart. Any change or update to the assets are reflected in the opportunity based renewal quote.



Please check out the **Must Knows** listed in the section Data Flow before setting up the data.

To get started with configuring data required to trigger renewal opportunity creation, please refer Setup and Configuration.

Points to Note

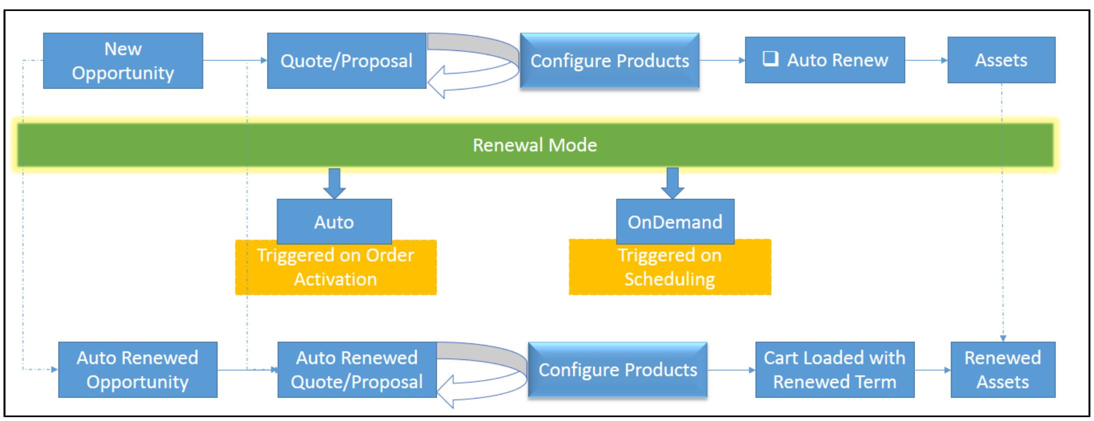
Before you start readying the system for renewal opportunity creation, you must know and remember the following:

- You must always enter an **Expected Start Date** and **Expected End Date** on the original Quote.
- If your Opportunity Quote has some mandatory fields, you must copy those field values using a Trigger.
- For a product, you must turn ON the **Auto Renew** flag either for *all* Charge types or *none*. You cannot apply different renewal approaches to different charge types of a product.
- For a Bundle product, you must turn ON the **Auto Renew** flag either for *all* the options within a bundle or none. You cannot split a bundle into two opportunities.
- When you set the **Auto Renew** flag to *true* for Renewal Mode as Auto, Opportunities will be created for all the Lines except Evergreen.
- Any changes to Assets will not be reflected and synchronized on the Opportunity.

- In the current version, you will not receive an Email notification on the generation of a Renewal Quote.
- Related Lists such as Activity and Related Proposals are not populated on the Opportunity.

Use-case

1. Specify a **Renewal Mode** and other related fields for renewal from Configuration.
2. Confirm that **Auto Renew** flag is set to true on **PLI** for the products you want to renew.
3. Create a Quote from an Opportunity and Configure Products. Add products to the cart and finalize it.
4. After **Presenting** and **Accepting** the Quote, activate the Order.
   The Assets are active and visible on your Account.
5. If **Renewal Mode** is set to *Auto*, Renewed Opportunity creation is triggered as soon as the Order gets activated.
   For the first time, you need to schedule a batch job *AssetRenewalJobScheduler* for auto-renewal.
   Go to **Setup** > **Develop**> **Apex Classes** and click **Schedule Apex**. Refer Setup and Configuration for details.
6. A renewed Opportunity and Quote is created. When you configure this quote, you will be taken to the Cart page where assets with renewed Start and End Dates are available.



Please note that on the new Renewal Quote, the **Start Date** = **Parent Quote's End Date** + 1 and **End Date** = **New Start Date** + Selling Term. However, you might see an incorrect End Date on a couple of occasions.

Get started with creating Renewal Opportunities; refer Opportunity Renewal - Auto or Opportunity Renewal - OnDemand renewal sections.

Setup and Configuration

Access **Config Settings** > **Installed Products Settings** to see a bunch of following Renewal Field Settings.

Field	Description
Renewal Business Object Type	Enter the Business object for which this renewal is taking place.
Renewal Default Price Book	Enter a Renewal Price Book name which you want to associate with renewals.
Renewal Execution Mode	Indicates if the Renewal of Asset Line Items must happen automatically or as per your specified conditions. • Enter **Auto** to automatically create the Renewal Opportunity on Order activation. Make sure you check **Auto Renew** on the product **PLI**, from the **Default** tab. • Enter **OnDemand** to create the renewal Opportunity before a certain lead time. You must enter a **Renewal Lead Time** for this mode to work successfully. To create a renewal Opportunity 90 days before the Asset Expiry, set the **Execution Mode** to *OnDemand* and **Renewal Lead Time** to *90*.

Field	Description
Renewal Group Fields	Indicates how to group the Asset Line Items when a Renewal Opportunity is created. Specify the API names of the fields you want to use for grouping. For Execution mode set to **Auto**, the system will group the renewal Opportunity by default by the **Auto Renew** Flag - One for the Asset Lines with **Auto Renew** as *True* and the other for Asset Lines with **Auto Renew** as *False*. For Execution mode set to **OnDemand**, the system can group the renewal Opportunity by Account and the Price List. However, if the implementation teams want to group the renewal opportunities by other parameters on the Asset Line Item, they can do so by specifying a comma separated list of API names of the fields in this section. It is recommended to limit the grouping to a maximum of 4 fields.
Renewal Lead Time	Indicates when the Renewal Opportunity will be created after an Order is activated. By default, the value is 0, which indicates that the Renewal Opportunity is created immediately after the Order is activated, with the same number of Line Items in the Order. If you specify 30, the Renewal Opportunity is created 30 days before the Asset End Date.
Default Renewal Price List	Specify the Name of the Price List which is a mandatory field for Quote creation.

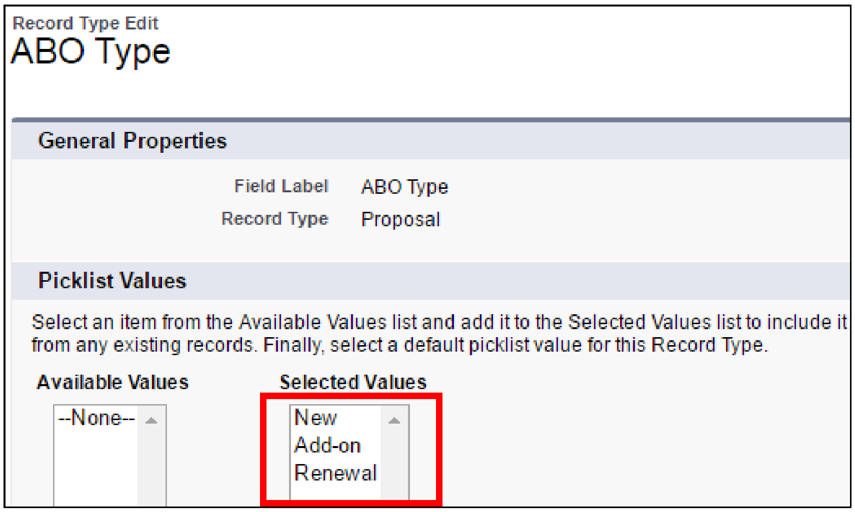
AutoSync Asset changes to an Opportunity based Renewal Quote

When you select the Sync Assets to Quote flag on a proposal, w henever an asset is changed or updated, the asset line item in the Opportunity based renewal quote is also updated.

- Asset based Orders sync does not support Swap and Cancellation actions.
- Blocking the Asset Based Order Synchronization should always be done on the renewal quote.
- Opting out of a renewal opportunity does not pull the line out from the auto renewal quote.

Add Picklist values to the Proposal Object

Go to the **Proposal** object and click to open Proposal from the **Record Types**.
Edit the Picklist **ABO Type** from the list and ensure you have added *New*, *Add-on* and *Renewal* to the list of **Selected Values**.



Renewal email Templates

To setup the email templates, navigate to **Setup** > **Communication Templates** > **Email Templates** and create templates for Asset Renewal Notification.

Select **Apttus CPQ Email Templates** from the **Folder** picklist to access existing templates or create a new one.

In the current version, you will not receive an Email notification on the generation of a Renewal Quote.

Batch job for Auto Renewal

You must schedule the batch job *AssetRenewalJobScheduler* for the first time. This job runs every 5 minutes after you have scheduled it.
Go to **Setup** > **Develop**> **Apex Classes** and click **Schedule Apex**. Enter a **Job Name** and select *AssetRenewalJobScheduler* as the **Apex Class**. Enter the **Start** and **End Date** and **Save** the schedule.

To monitor the job, go to **Setup** > **Apex Jobs** (under Monitoring), to see *AssetRenewalJobScheduler* in the list of jobs. You will get an email notification after this job is successfully completed.

Opportunity Renewal - Auto

For renewal subscriptions, you can automatically renew an existing Opportunity as soon as it gets Activated.

There are two ways in which renewal Opportunity is created for Auto renew and non-auto renew Line Items:

- Renewals with Filter
- Renewals without Filter

Renewals with Filter

You can take any field on the Asset Line Item and use it as a grouping field when creating a renewal Opportunity. In this section, you can see how to

- Group Renewal Opportunity by Product
- Group Renewal Opportunity by Product and Auto Renew Flag

Grouping Renewal Opportunity by Product

Let us take an example of renewing an Opportunity which has both auto renew and non-auto renew lines.

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	One-time	10000	12	4/1 /2015	3/31 /2016	1	10,000	24	10
Product A	Recurring	100	12	1/1 /2015	12/31 /2015	1	10,000	24	
Product B	One-time	5000	12	5/1 /2015	4/30 /2016	1	10,000	24	10

Product B	Recurring	200	12	1/1 /2015	12/31 /2015	1	10,000	24	

Configuration

To group the Renewal Opportunity by the Product, enter the *API name* of the **Product Name** field in **Renewal Group Fields**.

Field	
Renewal Execution Mode	Auto
Renewal Lead Time	-
Renewal Group Fields	Apttus_Config2__ProductId__r.Name

Opportunities Created

Opportunity 1

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	One-time	10000	12	4/1 /2015	3/31 /2016	1	10,000	24	10
Product A	Recurring	100	12	1/1 /2015	12/31 /2015	1	10,000	24	

Opportunity 2

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product B	One-time	5000	12	5/1 /2015	4/30 /2016	1	10,000	24	10
Product B	Recurring	200	12	1/1 /2015	12/31 /2015	1	10,000	24	

Grouping Renewal Opportunity by Product and Auto Renew Flag

This is the Opportunity that will undergo renewal.

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	One-time	10000	12	4/1 /2015	3/31 /2016	1	10,000	24	10
Product A	Recurring	100	12	1/1 /2015	12/31 /2015	1	10,000	24	
Product B	One-time	5000	12	5/1 /2015	4/30 /2016	1	10,000	24	10
Product B	Recurring	200	12	1/1 /2015	12/31 /2015	1	10,000	24	

Configuration

To group the Renewal Opportunity by the Product and Auto Renew Status, enter the *API name* of these fields in **Renewal Group Fields**.

Field	
Renewal Execution Mode	Auto
Renewal Lead Time	-
Renewal Group Fields	Apttus_Config2__ProductId__r.Name, Apttus_Config2__AutoRenew

Opportunities Created

Opportunity 1 Product A: auto renew

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustmen Amount
Product A	One-time	10000	12	4/1 /2015	3/31 /2016	1	10,000	24	10

Opportunity 2 Product B: auto renew

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product B	Recurring	200	12	1/1 /2015	12/31 /2015	1	10,000	24	

Opportunity 3 Product A: non-auto renew

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
	Recurring	100	12			1	10,000	24	

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A				1/1 /2015	12/31 /2015				

Opportunity 4 Product B: non-auto renew

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustment Amount
Product B	One-time	5000	12	5/1 /2015	4/30 /2016	1	10,000	24	10

Renewals without Filters

Let us take an example of renewing an Opportunity which has both auto renew and non-auto renew lines.

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	One-time	10000	12	4/1 /2015	3/31 /2016	1	10,000	24	10
Product A	Recurring	100	12	1/1 /2015	12/31 /2015	1	10,000	24	
Product B	One-time	5000	12	5/1 /2015	4/30 /2016	1	10,000	24	10
Product B	Recurring	200	12	1/1 /2015	12/31 /2015	1	10,000	24	

Configuration

Field	
Renewal Execution Mode	Auto
Renewal Lead Time	-
Renewal Group Fields	-

Opportunities Created

All non-auto renew

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	Recurring	100	12	1/1 /2015	12/31 /2015	1	10,000	24	
Product B	One-time	5000	12	5/1 /2015	4/30 /2016	1	10,000	24	10

All auto renew

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	One-time	10000	12	4/1 /2015	3/31 /2016	1	10,000	24	10
Product B	Recurring	200	12	1/1 /2015	12/31 /2015	1	10,000	24	

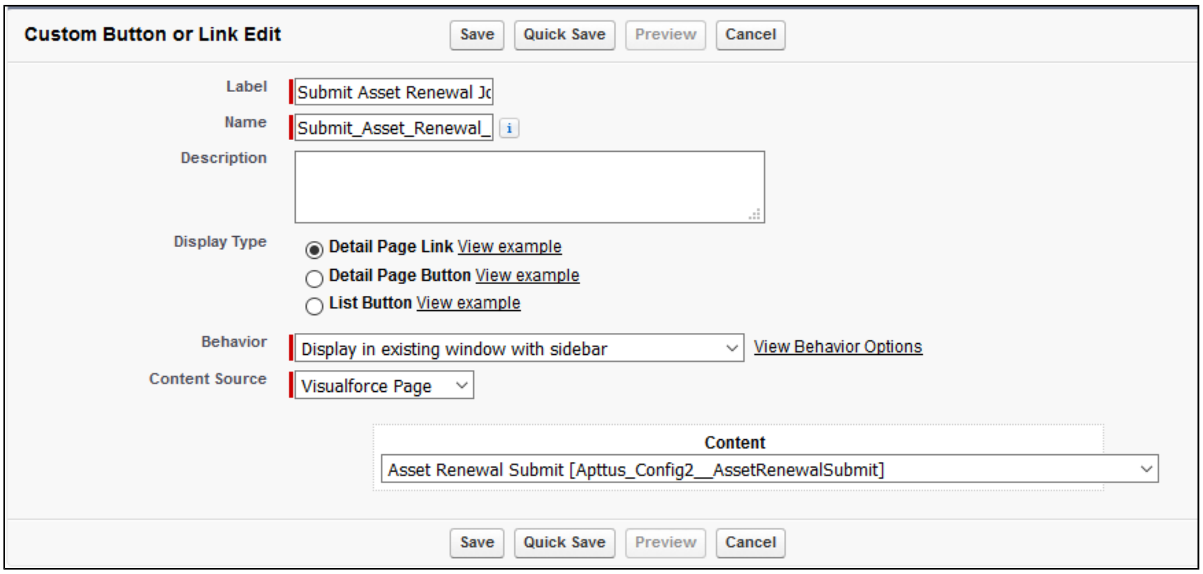
Opportunity Renewal - OnDemand

You can renew an Opportunity for assets that are due for expiry before a certain Lead Time. This mode of renewal is OnDemand Opportunity Renewal.

When you set **Renewal Execution Mode** as *OnDemand*, you must specify a **Renewal Lead Time** on the Installed Product Settings. For details please refer, Setup and Configuration.

Pre-requisite

- Navigate to Quote/Proposal and create a new link, set the content source as VF page and select *Apttus_Config2__AssetRenewalSubmit* as the VF page and click **Save**.



- To do this for multiple accounts, create a custom controller and VF page.

  Enter the following code in the custom controller. Create a reference for the custom controller in the VF page.

```
List<ID> accountIds =new List<ID>();

// gather account ids

// instantiate the OOTB controller and pass in the account ids

Apttus_Config2.AssetRenewalSubmitController baseController = new
 Apttus_Config2.AssetRenewalSubmitController(accountIds);
```



```
// submit the job

ID jobId = baseController.doSubmitJob();
```

OnDemand Auto Renewal without Grouping

Let us take an example of renewing an Opportunity which has both auto renew and non-auto renew lines.

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	One-time	10000	12	4/1 /2015	3/31 /2016	1	10,000	24	10
Product A	Recurring	100	12	1/1 /2015	12/31 /2015	1	10,000	24	
Product B	One-time	5000	12	5/1 /2015	4/30 /2016	1	10,000	24	10
Product B	Recurring	200	12	1/1 /2015	12/31 /2015	1	10,000	24	
Product C	One-time			7/1 /2015	6/30 /2016	1	10,000	24	
Product C	Recurring			7/1 /2015	6/30 /2016	1	1200	24	10

Configuration

Field	
Renewal Execution Mode	OnDemand
Renewal Lead Time	120
Renewal Group Fields	-

Opportunities Created

Non-auto renew (All assets expiring within 120 days from 1st Dec 2016)

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	Recurring	100	12	1/1 /2016	12/31 /2018	1	10,000	24	
Product B	One-time	5000	12	5/1 /2016	4/30 /2018	1	10,000	24	10

Auto Renew (note product C is not in the list)

Product	Charge Type	Base Price	Term	Start Date	End Date	Quantity	Net Price	Auto Renewal Term	Renewal Adjustme Amount
Product A	One-time	10000	12	4/1 /2016	3/31 /2018	1	10,000	24	10
Product B	Recurring	200	12	1/1 /2016	12/31 /2018	1	10,000	24	

Asset Renewal Callback Class

Using the Asset Renewal Callback class, you can filter out the renewal quote assets that are created using the On Demand job. For example you can filter out assets by Product Name as a filter criteria. To create an asset renewal callback class follow these steps:

1. Navigate to Apex Classes and click **New**.
2. Create a new Apex Class. Paste the code as shown in the sample callback class below. You can edit the filter expression specified in the ***getAssetFilterExpr()*** method to include your own customized filter criteria for the assets. In the sample below, the assets on the renewal quote are filtered based on the product name.

   Sample AssetRenewal Callback class

```
global with sharing class TestAssetRenewalCustomCallback
implements Apttus_Config2.CustomClass.
IAssetRenewalCustomCallback {



    /**

     * Callback invoked to append asset filter expression to
the renew asset line item query

     * Filter expression should be created using asset line
item fields only.

     * @return the filter expression or null to use the default
filter.

     */

    global String getAssetFilterExpr() {

        Product2 prod = [SELECT ID from Product2 where Name = 'S
tandalone2'];

        return 'Apttus_Config2__ProductId__c = \'' + prod.Id + '
\'';

    }
```



```
    /**

     * Gets the list of configuration properties for the given
business SObject

     * @param bObjectSO the business Sobject to get the
configuration properties for

     * @return the list of configuration properties.
     */

    global List<Apttus_Config2.Property> getConfigProperties
(SObject bObjectSO) {

        List<Apttus_Config2.Property> configProps = new
List<Apttus_Config2.Property>();



        configProps.add(new Apttus_Config2.Property('flow', 'LAn
gFlow'));

        configProps.add(new Apttus_Config2.Property('useAdvanced
Approval', 'false'));

        configProps.add(new Apttus_Config2.Property('useDealOpti
mizer', 'false'));



        return configProps;

    }



}
```

3. Specify a name for the callback class and click Save.
4. Navigate to **Develop > Custom Settings > Config Custom Classes > Manage**.
5. You can do any one of the following:
   - If a custom class setting exists, click Edit.

- To create a new Custom Class, click New and specify the name as Custom Classes.

6. In the Asset Renewal Custom Callback Class field specify the name of the class you have created.
7. Click **Save**.
   When an On-Demand renewal quote is created, the assets on the renewal quote are the ones that are fetched by the ***getAssetFilterExpr()*** method.

Setting up Asset-based Pricing without Workflow rules

Using a workflow rule to obtain the price of an asset during Asset-based operations has certain limitations. You could use workflow rules to fetch asset pricing on quantity and term updates but the process gets complicated when it involves attribute or other field changes.
Asset pricing without workflow rules helps you pick the selling price of an asset or its new price from the Price List item as and when needed.

With this feature, you can choose to

- Retain the asset price on the cart (default asset price)
- Go back to the Price List Item and fetch the new price of the asset. This happens when the parameters driving pricing undergo change.

Configuration

To turn ON Asset-based pricing without creating a workflow rule, you must configure the following two parameters:

- **Default Asset Pricing Indicator**: When checked, the system loads the asset price by default in the cart and monitors the changes done to fields added in the Asset Pricing Criteria.
  Go to **Custom Settings** > **Config System Properties** to turn ON the pricing indicator.
- **Criteria Field Names**: Enter the API name of the field like Quantity or attribute that drives pricing.
  Go to **Custom Settings** > **Config Asset Pricing Criteria Fields** to provide the field APIs. Any changes to the fields specified here will trigger the flip and you will see the price from PLI on the cart. If you revert the update, the price will change to the default asset price.

> ⓘ
> - If the default asset pricing indicator is off, then no asset pricing criteria should be defined.

- Asset pricing is always turned on if there asset pricing criteria defined.
- Asset pricing indicator is at a global level. Any requirements to turn it off on a selective basis must be handled through a workflow rule to deselect the indicator on the cart load.

Use-Case

1. Create a product ‘B’ with tiered pricing. The per unit price of the product is $1200.
2. Ensure **Default Asset Pricing Indicator** is set to True.
3. Create a Quote with one unit of product ‘B’ for 1200. Offer a discount of $200 to see that the price of asset ‘B’ now becomes $1000.
4. Enter the API name of the field **Quantity** under the **Criteria Field Name**.
5. From the CSR flow, select this asset ‘B’ and perform the ABO operation of ‘**Change**’. You’ll see the asset price of $1000 on the cart.
6. Change the **Quantity** to *‘5’* and you will see that asset price is now set to $1200 (from the PLI).

Standalone Asset - Quantity Update

Similar to the above use-case, the parameter **Quantity** drives pricing in the following case.

Standalone Asset	Asset pricing Enabled	Base Price	Quantity	Term	Extended Price	Discount	Net Price	Asset Ne Price (Net pric /Quantity
Initial Sale	N/A	100	2	12	2400	350	2050	85.416
On load of the Cart	True	85.416 (fetched from Asset Line Item)	2	12	2050		2050	
	True		4	12	4100	100	4000	83.333

Standalone Asset	Asset pricing Enabled	Base Price	Quantity	Term	Extended Price	Discount	Net Price	Asset Ne Price (Net pric /Quantity
On Change (asset)		85.416 (fetched from Asset Line Item)						

Standalone Asset - Quantity Update

The Attribute **Service Level type** Gold has the price $100 and type Platinum $200. The Asset price will change based on the changes made to Service Level attribute.

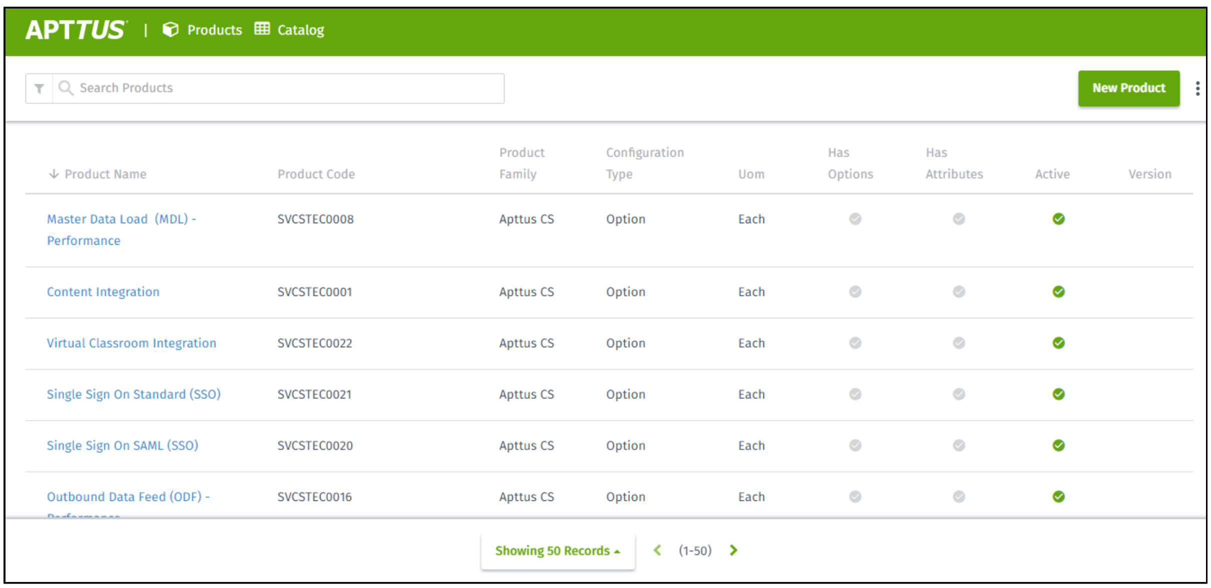
Standalone Asset	Attribute (Service Level)	Base Price	Quantity	Term	Extended Price	Discount	Net Price	Asset Ne Price (Net pric /Quantity
Initial Sale	Gold	100	2	12	2400	350	2050	85.416
On load of the Cart	Gold	85.416 (fetched from Asset Line Item)	2	12	2050		2050	
On Change (asset)	Platinum		2	12	4800	100	4700	195.8333

Standalone Asset	Attribute (Service Level)	Base Price	Quantity	Term	Extended Price	Discount	Net Price	Asset Ne Price (Net pric /Quantity
		200 (fetched from Price List Item)						

New CPQ Admin Console

With this feature, an admin can use the new administrative console to manage products and its related components such as options, attributes, and categories. The admin console helps you to create new product, associate the product with the categories, price lists and price list items on a single page.

You can navigate to **Apttus CPQ Admin** app, and click **CPQ Admin** tab to access the new admin console.



In the Product Search box, you can search for the product as per the product name to narrow down the product list. Clicking the icon opens the Filters section, which help you filter the products by their types.

Filter	Description
Recently Modified	To view the products which are recently modified.
Bundle	To view all bundle products.
Standalone	To view all standalone products.

Filter	Description
Options	To view all option products.

At the bottom of the page, you can control the number of products to be displayed per page. By default, you see 50 records per page.

You can configure the fields that are displayed on this screen from **Setup > Customize > Products > Field Sets > Product List Fields.**

To configure the CPQ Admin landing page fields

1. Navigate to **Setup > Customize > Products > Field Sets > Product List Fields.** This field set displays the fields on the landing page of CPQ Admin.
2. Click **Edit** for **Product List Fields**
3. Drag and drop the required fields in the **In the Field Set**.
4. Click **Save**.

Click **New Product** to create a new product configuration.

You can configure the fields that are displayed on the screen from **Setup > Customize > Products > Field Sets > Produc Detail Fields.**

To configure the Product creation screen fields

1. Navigate to **Setup > Customize > Products > Field Sets > Product Detail Fields.** This field set displays the fields on the product creation screen of CPQ Admin.
2. Click **Edit** for **Product Detail Fields.**
3. Drag and drop the required fields in the **In the Field Set**.
4. Click **Save**.

When you click **New Product**, you land on the Product Creation screen.

Once you are done, you can click **Save Product**. Click **Cancel** to roll back the product creation and go back to the previous page.

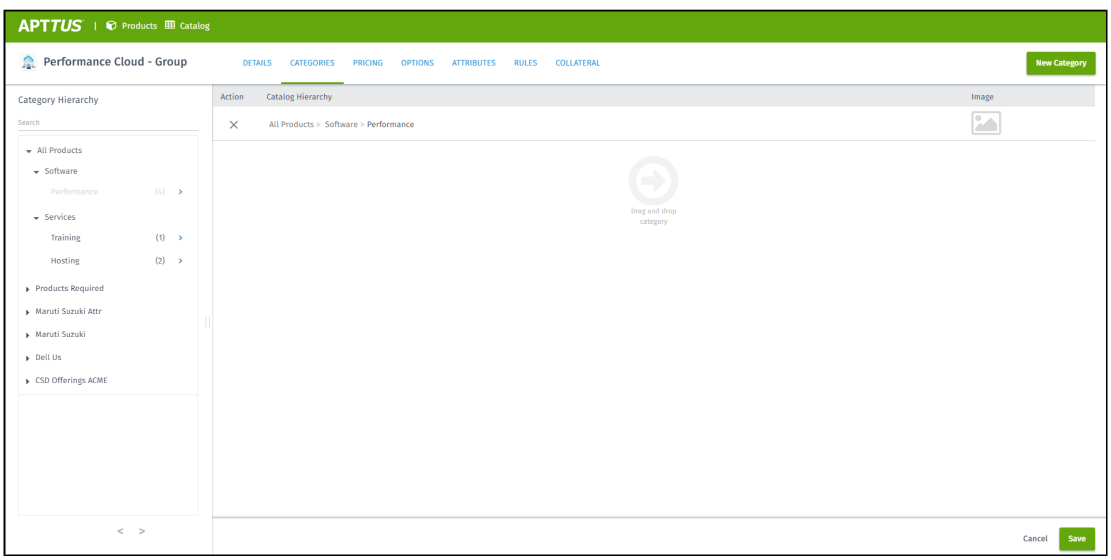
There are multiple tabs for each product that you can use to associate the categories, create attributes, price list items, and more. As you navigate from one tab to the other, system saves the data that you enter in the previous screen.

Details Section

In the Details tab, you can specify the product specific details as per the field sets that you have defined. You can choose to modify different fields such as UOM, Expiration Date, Has Attributes, Has Options and more.

Categories Section

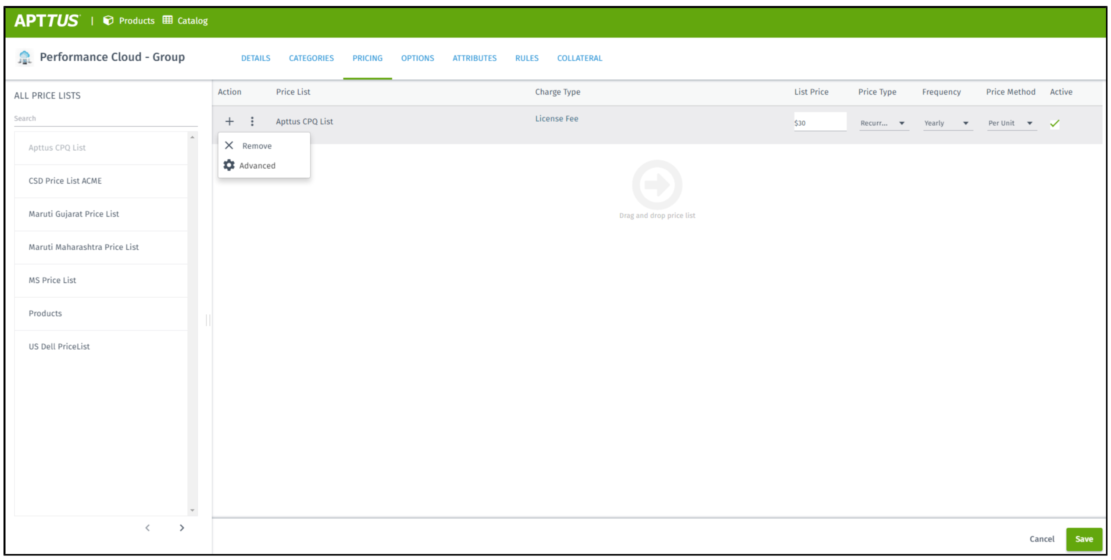
In the Categories tab, you can drag and drop the desired categories to which you want the product to be associated with. You can search for the category through Category Hierarchy search box. Once you have associated the category, such category is greyed out in the Category Hierarchy section on the right hand side. Category Hierarchy also shows the number of products that belong to the category along with its name.



Pricing Section

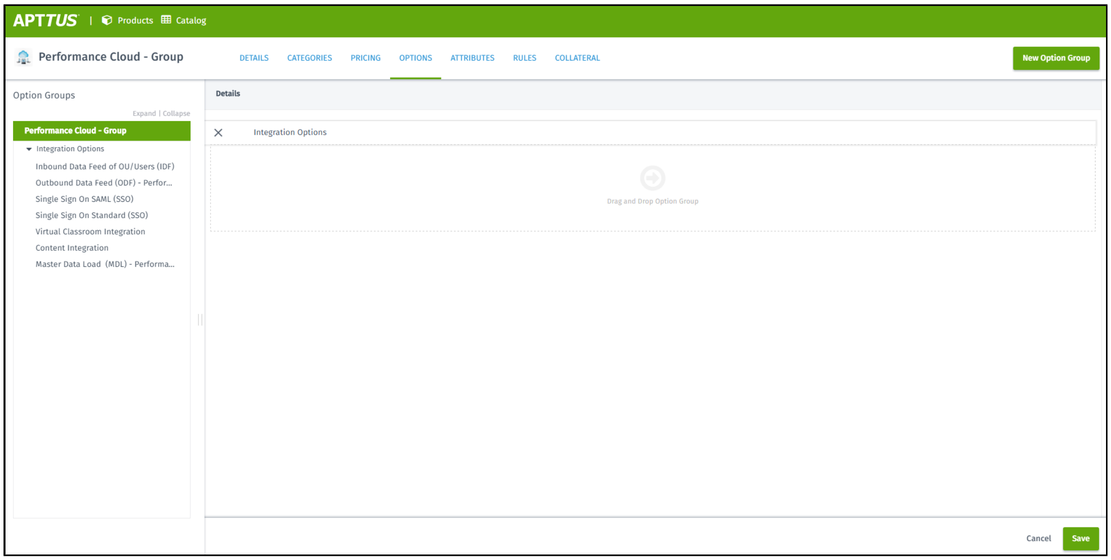
In the Pricing tab, you can associate the price list(s) to the product. You can also assign the Charge Type, Sequence, List Price, Price Type, Frequency, Price Method, Effective date, Expiration date. Select **Active** check box to activate this Price List. For a single product, if you want to create different Price List Items supporting different Charge Types, you can click corresponding icon of the Price List Item.

Click **Add New** to create another Price List Item.



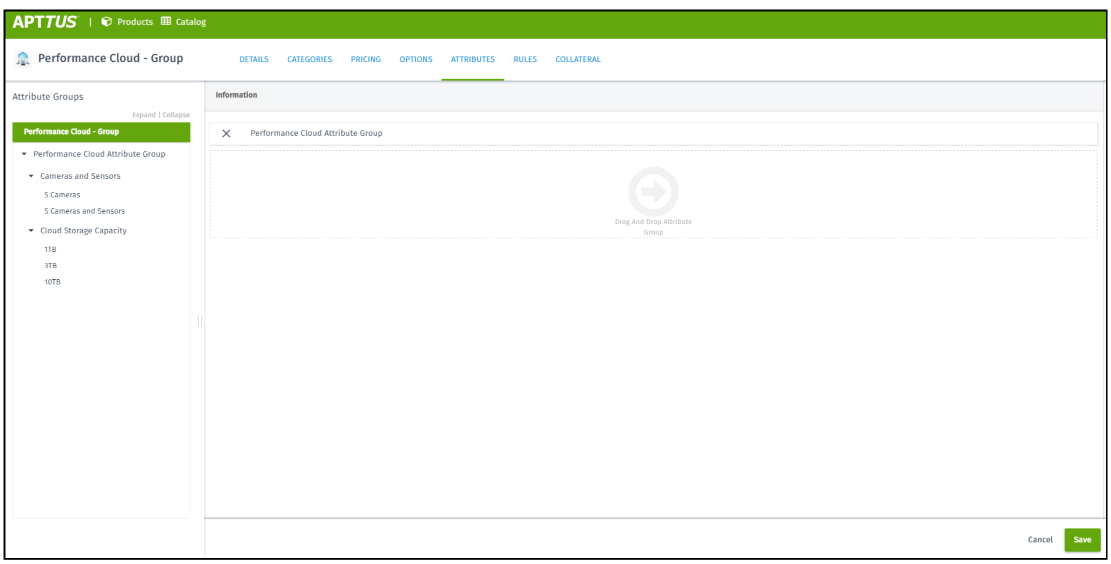
Options Section

For Bundle and Option product which is marked as Configuration Type = Bundle or Configuration Type = Option, you see additional tab - Options. In the Options tab, you can drag and drop the desired option group and options that you want the product to be associated with. You can configure the option group by clicking on the Option Group name once you have associated the option group with the product.

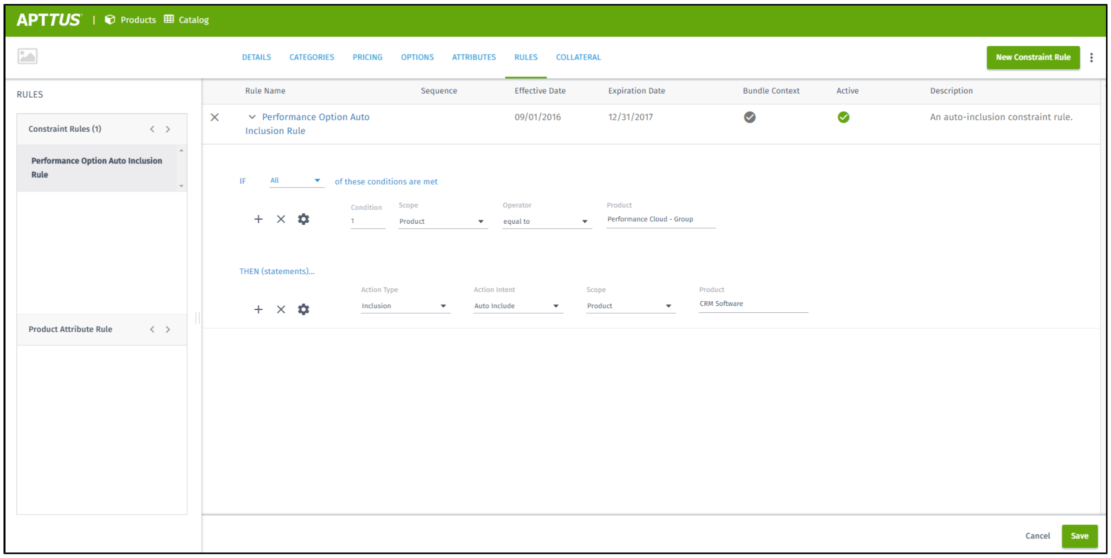


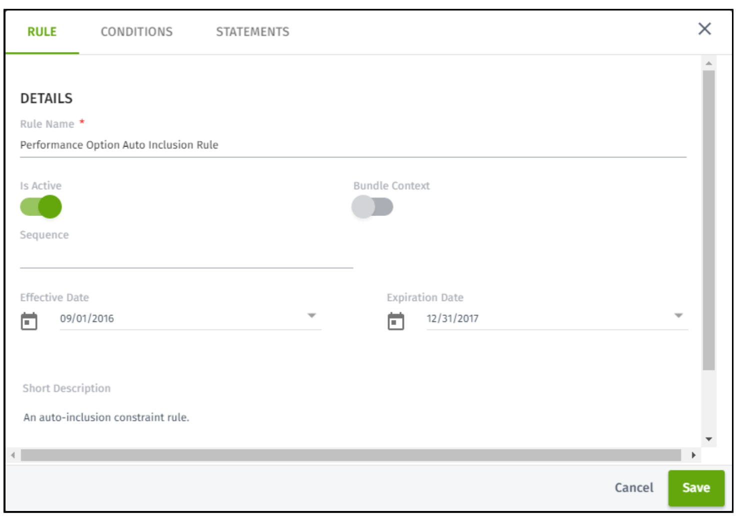
Attributes Section

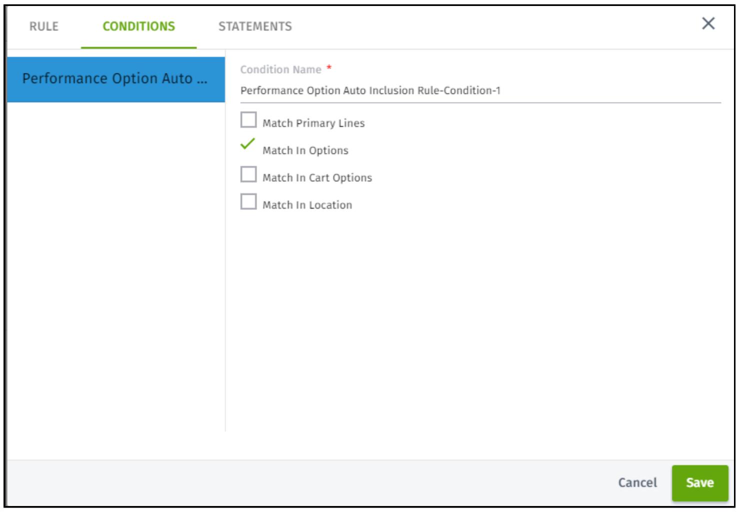
In the Attributes tab, you can drag and drop the desired attributes to which you want the product to be associated with. You can search for the attribute through Attribute Group search box. Once you have associated the attribute, such attribute is greyed out in the Attribute Group section on the right hand side.

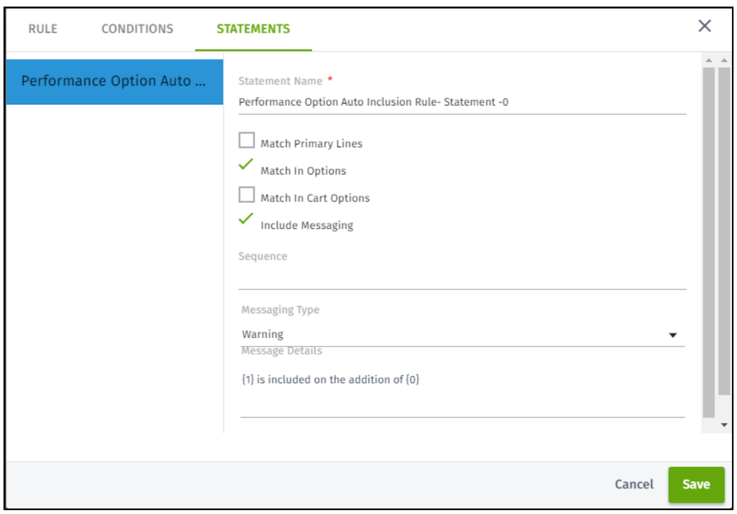


Rules Section



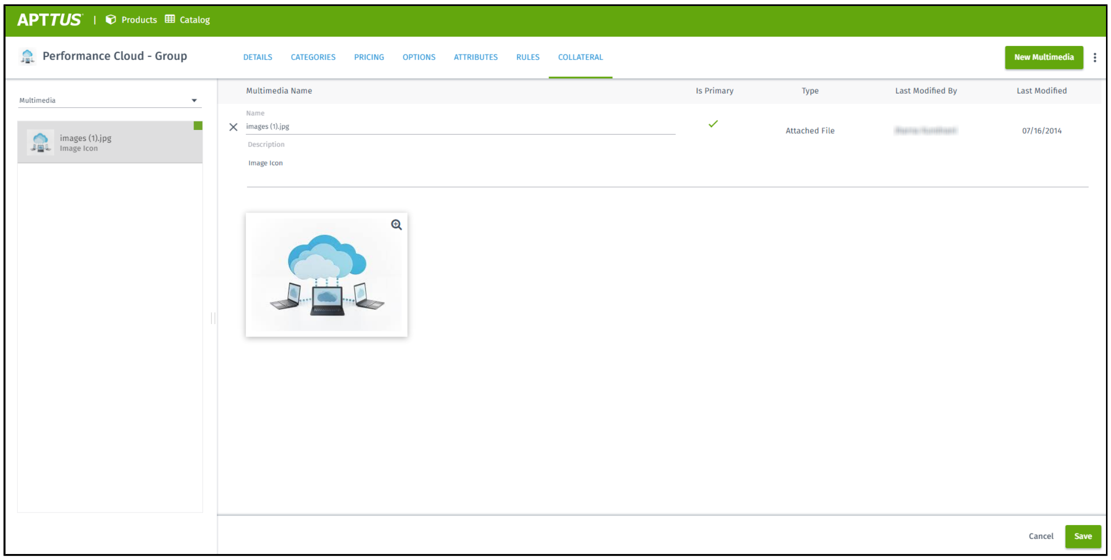






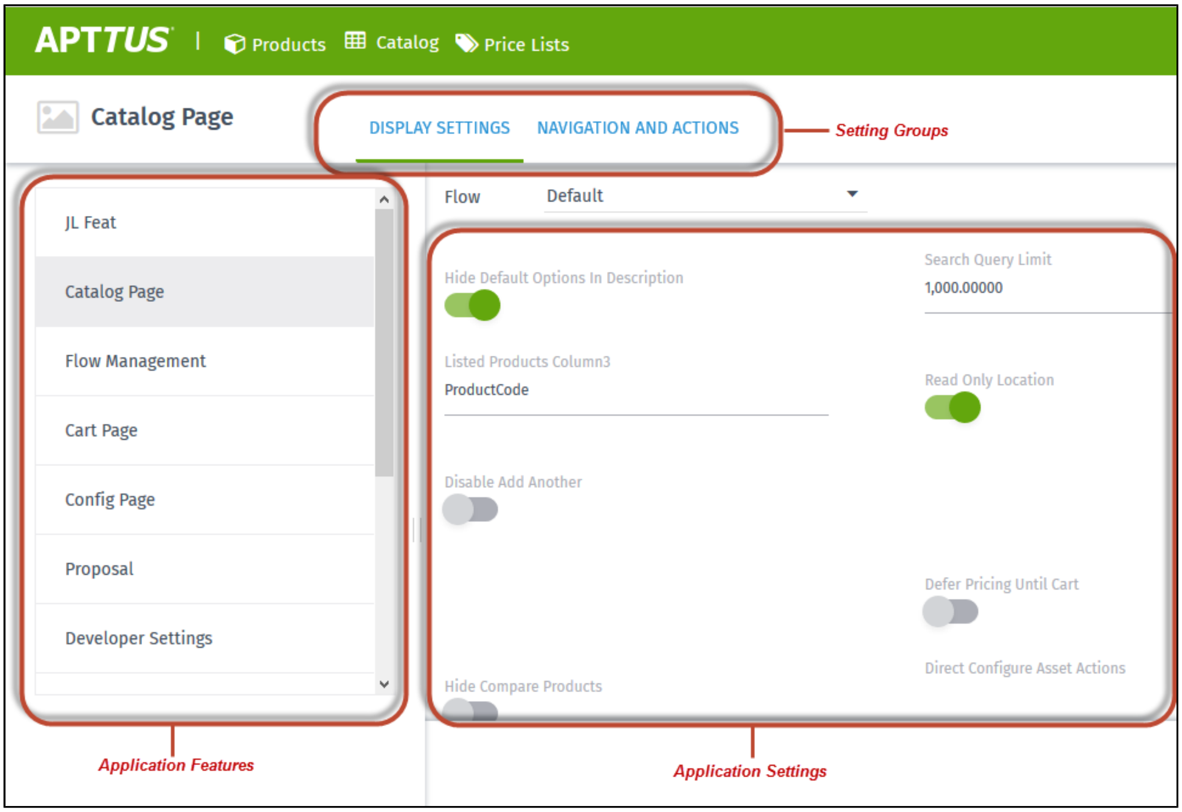
Collateral Section

You can associate multiple images to the product, from **Image(s)** section and clicking on **Upload** button.



System Settings Page

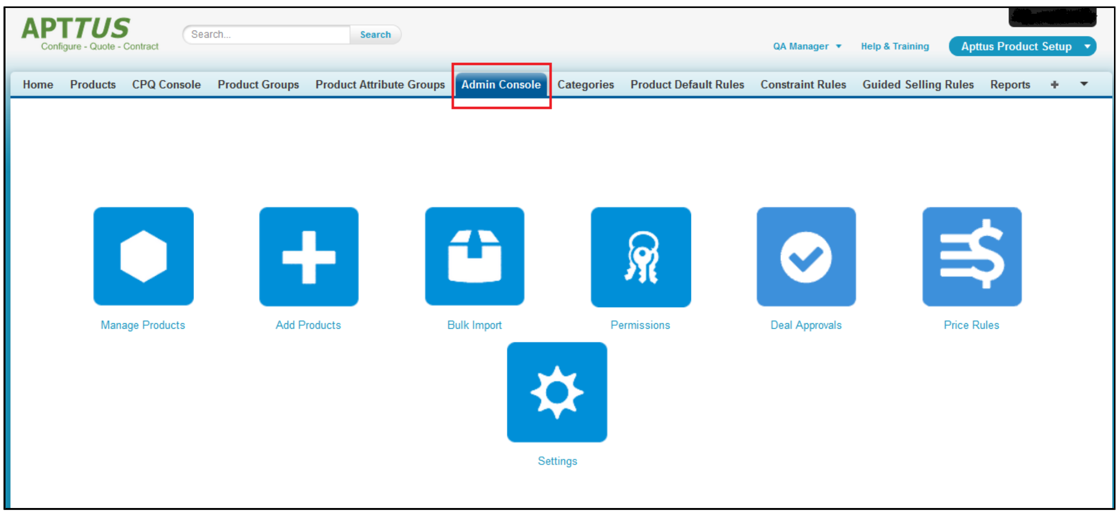
Go to **Admin Console** and click **System**. All the settings created using the Application Settings Management pages appear here. You can set and edit the Application Feature Settings on this page.


APTTUS
Products
Catalog
Price Lists
Catalog Page
DISPLAY SETTINGS
NAVIGATION AND ACTIONS
Setting Groups
JL Feat
Catalog Page
Flow Management
Cart Page
Config Page
Proposal
Developer Settings
Flow
Default
Hide Default Options In Description
Search Query Limit
1,000.00000
Listed Products Column3
ProductCode
Read Only Location
Disable Add Another
Defer Pricing Until Cart
Direct Configure Asset Actions
Hide Compare Products
Application Features
Application Settings

Sprint CPQ

Accessing the Admin Console

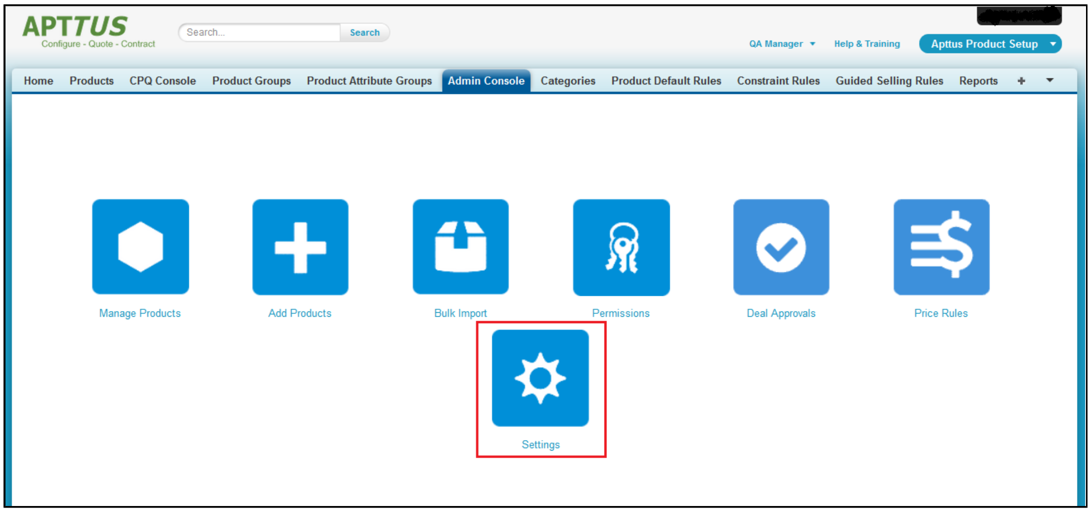
With the new release, Sprint CPQ is available in the Configuration & Pricing package and you can navigate to the Sprint CPQ Console with a new tab, named **Admin Console**. Clicking this tab allows you to perform the same functions as Sprint CPQ Console, such as Adding and Managing Products, Price Rules, Permissions, and more.



Accessing Config Settings

Now, the administrator functions are more accessible on the Sprint CPQ Console. You can directly go to the Config Settings page from the **Admin Console**.

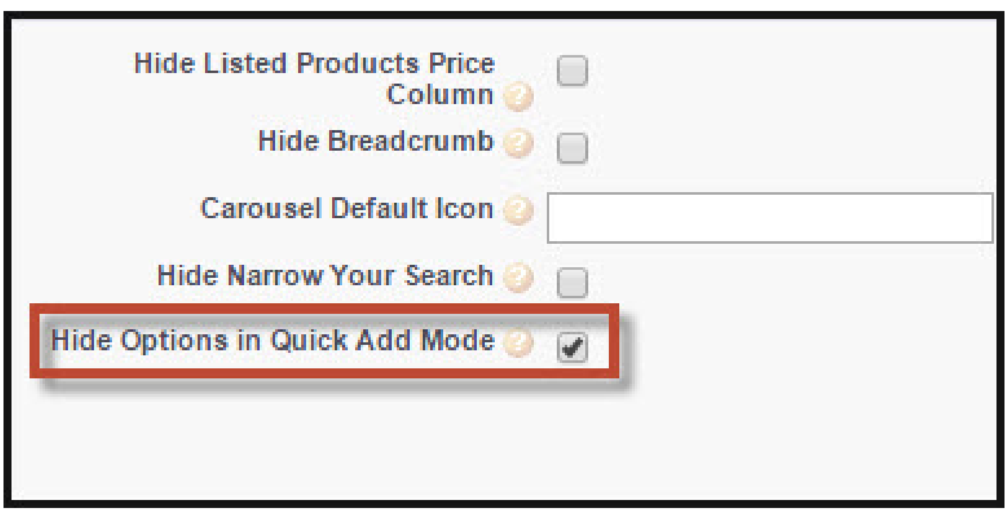
Simply click **Settings** and on the top right corner of the Manage Application Settings page, click **Advanced** and you can edit the Config Settings.



Hiding options from QuickAdd page

Now, you can hide those bundle options from the product catalog page, that are not compatible with the customer's original purchase.

When you browse the Quick add page, you can see several bundle options displayed in the catalog. Many of these options are not associated with your customer's original purchase. Sales representatives in your organization could include such bundle options that do not meet your customer's requirements or could be of little or no value to them. To prevent returns, exchanges, and cancellations, you must hide such bundle options from the product catalog page.



To enable the Hide Options in Quick Add Mode feature,

1. Click **Config Settings > Catalog Page settings**.
2. In the Product List Settings section, select the **Hide Options in Quick Add Mode** check box.
3. Click **Save**.

Configuring Bundle Products on the Quick Add page

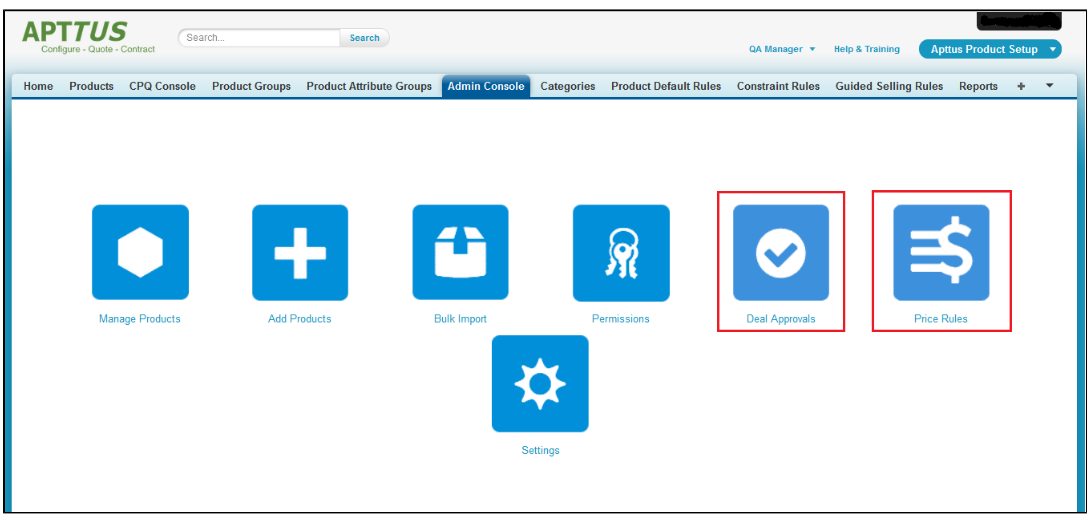
You can now configure a Bundle Product on the Quick Add or Catalog page.

Sprint CPQ now has the **Configure** button next to each bundle product listed on the catalog page.

To configure a bundle product on the Quick Add page, click the **Configure** button.

Accessing Deal Approvals and Price Rule-sets from the Admin Console

You can now access Deal Approvals and Price Rules from the Admin Console.



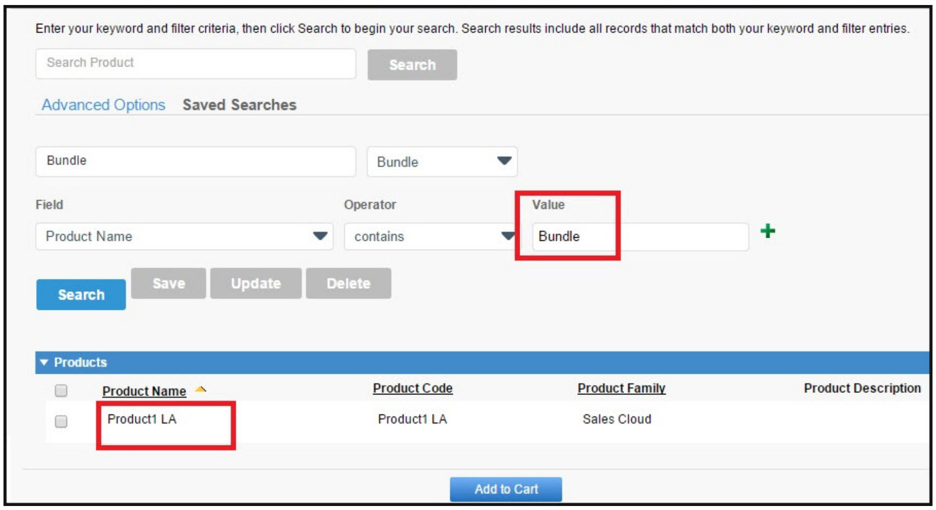
Click **Price Rules** to see the list of Price Rule-sets and click **Deal Approvals** to see the Approval Process page.

Note

Now, the CartDetailView page displays the shopping cart in Sprint CPQ.

Saving your Searches

With Sprint CPQ, you can now save your searches This feature allows you to save the products and their criteria you searched for the Search dialog box. You can also update and delete your saved searches.



To configure the Search on the Catalog Page,

1. Select an appropriate quote and click **Configure Products**.
2. Click **Saved Searches**.
3. Select **Field**, **Operator,** and **Value**.
4. Click **Search** to search for the products.

Click **Save New** to save the search.

Apttus Configuration & Pricing Package Objects

Apttus CPQ is comprised of multiple packages as mentioned in the installation section and this page lists out the objects inside each package.

Click the right arrow key or the title to view the expanded list of objects per package.

Apttus Configuration and Pricing

Apttus CPQ Object	Purpose of the Object	Available in Versions
Account Billing Summary	Represents the billing summary for an account.	7.707 - Current
Account Location	Represents a location for an account.	6.674 - Current
Adjustment Line Item	Represents an adjustment to the price of a product/service.	6.674 - Current
Agreement Price Rule	Represents a special pricing rule associated with an agreement or an asset.	6.674 - Current
Agreement Price Tier	Represents a single entry in an agreement price tier.	6.674 - Current
Application Feature	An object to contain features in the Applications, where Applications are Apttus products like CPQ, CLM etc.	9.1157- Current
Application Setting	Object to contain metadata about application settings.	

Apttus CPQ Object	**Purpose of the Object**	**Available in Versions**
		9.1157-Current
Applied Expression Info	Represents expressions applied to the configuration.	8.766 - Current
Applied Rule Action Info	Rule actions that are applied to the configuration.	6.674 - Current
Applied Rule Info	Keeps track of rules applied to a configuration.	6.674 - Current
Asset Attribute Value	Represents the attribute values for a product class.	6.674 - Current
Asset Attribute Value History	Represents the attribute values for a product class.	9.933 - Current
Asset Line Item	Represents a product or service asset line item.	6.674 - Current
Asset Line Item History	Represents a product or service asset line item history.	8.827-Current
Asset Transaction History	Represents a transaction that records the change to an asset.	9.933 - Current
Asset Usage Price Tier	Represents a single entry in an asset usage price tier.	6.674 - Current
Asset Usage Price Tier History	Represents a single entry in an asset usage price tier.	9.933 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Attribute Group Translation	This object holds the translated values of attribute group object fields for all the languages.	9.1121-Current
Attribute Value Matrix	Represents the information about attribute value compatibilities or incompatibilities.	9.898-Current
Attribute Value Matrix Entry	Represents a row (entry) in an attribute value matrix.	9.898-Current
Batch Job (CPQ)	Represents an individual batch job.	6.674 - Current
Batch Job (CPQ) Parameter	Represents a batch job bind parameter.	9.977-Current
Billing Plan	Represents a billing plan associated with a product or service.	7.729 - Current
Billing Plan Item	Represents a billing entry in a billing plan.	7.729 - Current
Billing Plan Template	Represents the template to create a billing plan.	9.1118-Current
Billing Plan Template Item	Billing Plan Template Item.	9.1118-Current
Billing Preference	Represents a functional preference for billing purposes.	7.707 - Current
Bundle Component View	Keeps the list references of component products related to the bundle and sub-bundle.	6.674 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Cart Clone Info		9.1105 - Current
Category	Represents a category.	6.674 - Current
Category Hierarchy	Represents a category hierarchy.	6.674 - Current
CategoryTranslation	Catalog Hierarchy field translations are stored in this object. Each Category Translation object holds a translation for a SFDC supported language for a defined Category Hierarchy Object record.	9.1123- Current
Charge Group	Represents a group of charge types associated with a product or service.	7.707 - Current
Charge Group Member	Represents the mapping between a charge group and the charge types associated with the group.	7.707 - Current
Charge Type	Represents a type of charge for a product or service.	7.707 - Current
Collaboration Request	This object will persist information for quote collaboration tasks and will be used to relate the parent configurations and its related child configurations.	9.1040 - Current
Constraint Rule	Captures information about configuration constraint rule.	6.674 - Current
Constraint Rule Action	Captures information about rule actions. Rule actions are applied only when the rule conditions are satisfied.	6.674 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Constraint Rule Action Expression		8.822-Current
Constraint Rule Condition	Captures information about the condition of a constraint rule. In order for a rule to trigger all or some of the rule conditions should be true and the combining expression should be true.	6.674 - Current
Cost Adjustment	Represents an adjustment to the cost of a product/service.	9.1098 - Current
Cost Item	Represents a product entry in a Cost Model.	9.1157-Current
Cost Line Item	Represents cost associated with base price of line item.	9.1098 - Current
Cost Model	Cost Model represents cost model of a FES or Factory.	9.1098 - Current
Cost Type	Represents cost hierarchy definition.	9.1098 - Current
Custom Message	Used to provide translation for messages used in custom objects.	6.674 - Current
Document Collate Info	Object to contain information about all collated documents.	8.111 - Current
External Order Adjustment Item	Represents an adjustment to the price of a product/service.	9.990-Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
External Order Summary	Represents an external order summary associated with an account.	9.990-Current
External Order Summary Item	Represents a product or service line item associated with an external order summary.	9.990-Current
Favorite Configuration		9.1098 - Current
Feature	Represents a single feature in a feature set.	6.674 - Current
Feature Set	Represents a set of features shared by products.	6.674 - Current
Field Expression	Represents a field expression.	8.766 - Current
Footnote	Represents a single footnote in a document.	6.674 - Current
Formula Field (CPQ)	Represents a formula field used in price dimensions and rules.	6.674 - Current
Frequency Conversion Rate	Represents the conversion rates used to convert pricing from one frequency to another.	6.674 - Current
G/L Account Map Entry	Represents a single entry in a G/L account map rule.	7.717 - Current
G/L Account Map Rule	Represents a G/L account map rule to determine G/L account numbers for a product or service.	7.717 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Guided Question	Represents a question in a guided interview.	6.674 - Current
Guided Search Rule Entry	Represents a single entry in a guided search rule.	6.674 - Current
Guided Search Rule Filter	Represents a single search filter in a guided search rule entry.	6.674 - Current
Guided Selling Rule	Represents a rule to search for products based on a guided interview.	6.674 - Current
Help Doc	Object holds help documentation.	9.934 - Current
Incentive	Represents an incentive associated with one or more products.	9.830- Current
Incentive Adjustment Item	Represents an adjustment to the price of a product/service.	9.1034 - Current
Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit.	9.924 - Current
Incentive Coupon	Represents a coupon associated with an incentive.	9.834- Current
Incentive Group	Represents a group of incentives.	9.836- Current
Incentive Limit	Represents a single limit associated with an incentive.	9.832- Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Incentive Limit Data	Represents a single limit data row associated with the incentive limit.	9.832- Current
Incentive Loyalty Enrollment	Represents a single customer enrollment into an incentive loyalty program.	9.1133- Current
Line Item	Represents a product or service line item.	6.674 - Current
Line Item Rollup	Holds rollup summary custom fields.	8.766 - Current
Loyalty Point Accrual Adjustment Item	Represents loyalty points accrual adjustments for a product /service.	9.1156- Current
Loyalty Point Accrual Summary	Represents a summary of loyalty points accrued for the account or contact.	9.1149- Current
Milestone	Represents milestones associated with other business objects.	9.928 - Current
Milestone Adjustment Item	Represents an adjustment to the price of a product/service.	9.1000- Current
Milestone Template	Represents a template for creating milestones.	9.1000- Current
Notification Feed	A generic object to contain Notification Feeds generated by any object for any page.	9.1054 - Current
Order	Represents an order associated with an account.	

Apttus CPQ Object	**Purpose of the Object**	**Available in Versions**
		7.707 - Current
Order Adjustment Line Item	Represents an adjustment to the price of a product/service.	9.832- Current
Order Fulfillment	Represents an order fulfillment associated with an account.	9.1100 - Current
Order Fulfillment Line Item	Represents a fulfillment line item associated with one or more orders.	9.1100 - Current
Order Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the order.	9.924 - Current
Order Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the order.	9.864- Current
Order Line Item	Represents a product or service line item associated with an order.	7.707 - Current
Order Loyalty Point Accrual Item	Represents loyalty points accrued for a product/service.	9.1149- Current
Order Pocket Adjustment Line Item	Represents a pocket price adjustment to the price of a product /service.	9.871- Current
Order Product Attribute Value	Represents the attribute values for a product class in an order line item.	7.707 - Current
Order Rollup Data	Represents the dynamic rollup data for use in criteria.	

Apttus CPQ Object	Purpose of the Object	Available in Versions
		9.903-Current
Order Tax Breakup	Represents the tax breakup of a product/service	8.781-Current
Order Usage Price Tier	Represents a single entry in an order usage price tier.	7.707 - Current
Payment Term	Represents a payment term associated with a quote, order or an agreement.	7.707 - Current
Permission Set Relationship		8.779-Current
Price Breakup	Represents the price breakup of a product/service.	6.674 - Current
Price Dimension	Represents a price criteria dimension.	6.674 - Current
Price List	Represents a price list that contains the list of products the organization sells.	6.674 - Current
Price List Category	Represents the categories associated with a price list.	6.674 - Current
Price List Item	Represents a product entry in a price list.	6.674 - Current
Price Matrix	Represents a price matrix associated with a price list entry.	6.674 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Price Matrix Entry	Represents a single entry in a price matrix.	6.674 - Current
Price Rule	Represents a single rule in a price ruleset.	6.674 - Current
Price Rule Entry	Represents a single entry in a price rule.	6.674 - Current
Price Ruleset	Represents a rule set that contains the list of rules to modify prices of products the organization sells.	6.674 - Current
Product Attribute	Represents a single attribute in a product attribute group.	6.674 - Current
Product Attribute Group	Represents a product attribute group that contains attributes shared by products.	6.674 - Current
Product Attribute Group Member	Represents the association between a product and an attribute group.	6.674 - Current
Product Attribute Matrix View	Represents a denormalized (materialized) view of product attribute and matrices.	9.898- Current
Product Attribute Rule	Represents the rule definition of a product attribute value.	9.898- Current
Product Attribute Rule Action	Represents a single action in a product attribute rule.	9.898- Current
Product Attribute Rule View	Represents a denormalized (materialized) view of product and attribute rule.	9.1167- Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Product Attribute Value	Represents the attribute values for a product class.	6.674 - Current
Product Classification	Represents a product classification which is a member of the list of products associated with a classification.	6.674 - Current
Product Configuration	Represents a product configuration.	6.674 - Current
Product Constraint	Represents a set of constraints a product must satisfy before it can be added to the product configuration.	6.674 - Current
Product Constraint Entry	Represents a single entry in a product constraint.	6.674 - Current
Product Constraint View	Creates product and constraint rule view table based on the definition of product scope in constraint rule condition.	6.674 - Current
Product Default Rule	Represents a rule to default products, quantity, and term.	6.674 - Current
Product Default Rule Filter	Represents a single filter in a product default rule.	6.674 - Current
Product Default Value	Represents the default values for a product.	6.674 - Current
Product Feature	Represents the association between a product and a feature set.	6.674 - Current
Product Feature Value	Represents the feature values for a product and its associated feature set.	6.674 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Product Filter View	Represents the filter values for each product in a category.	6.674 - Current
Product Footnote	Represents the footnote available in a product or option.	6.674 - Current
Product Group	Represents a group of products.	6.674 - Current
Product Group Member	Represents a single product in a product group.	6.674 - Current
Product Hierarchy View	Represents a denormalized (materialized) view of product and classification hierarchies.	6.674 - Current
Product Information	Represents information associated with a product or a classification.	6.674 - Current
Product Option Component	Represents a product option component which is a member of the list of options associated with a product option group.	6.674 - Current
Product Option Group	Represents a product option group which is a member of the list of option groups associated with a product.	6.674 - Current
Product Option Price	Represents the price for an option or component.	6.674 - Current
Product Translation	This object holds the translated values of product object fields for all the languages.	9.1121- Current
Published Favorite	A junction object between Favorite Configurations and Price Lists to support many to many relationships.	9.1129- Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Related Account Location	Holds the relationships between account locations.	6.674 - Current
Related Asset Line Item	Holds the relationships between asset line items.	7.718 - Current
Related Incentive	Holds the relationships between incentives.	9.834- Current
Related Price List Item	Holds the relationships between price list items.	6.674 - Current
Related Product	Related products are for up-sell, cross-sell or related products (search helper).	6.674 - Current
Revenue Recognition Policy	Represents a revenue recognition policy associated with a product or service.	7.717 - Current
Revenue Split Policy	Represents an order revenue split policy associated with a product or service.	9.1045 - Current
Revenue Split Policy Entry	Represents a single entry in a revenue split policy.	9.1045 - Current
Rollup Data	Represents the rollup data for use in criteria.	9.833- Current
Saved Search	Represents a saved search criteria.	9.845- Current
Search Attribute Value	Holds product search parameters and default values for line items.	6.674 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Search Filter (CPQ)	Represents a search filter.	6.674 - Current
Service Location	Represents a service location.	9.1034 - Current
Setting Group	Object to group settings into logical sets.	9.1157- Current
Smart Search Activity History	Contains a history of the actions occurring during the search process when connecting to the Azure system (activation, deactivation, sync).	9.1120- Current
Summary Group	Represents a summary of line items.	6.674 - Current
Tax Breakup	Represents the tax breakup of a product/service.	8.780- Current
Tax Certificate	Represents a tax certificate.	7.707 - Current
Tax Code	Represents a tax code associated with a product or service.	7.707 - Current
Temp Display Column	Holds information about columns displayed in the cart view.	6.674 - Current
Temp Filter	Helps in the binding of search filter values.	6.674 - Current

Apttus CPQ Object	Purpose of the Object	Available in Versions
Temp Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the product configuration.	9.924-Current
Temp Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the product configuration.	9.864-Current
Temp Object (CPQ)	A container for temporary objects used by the application.	6.674 - Current
Temp Renew	Temporary object for capturing user input to create asset line item during the amendment/renewal process.	6.674 - Current
Temp Renew Asset Group	Temporary object to store renew asset group processed by RenewOpportunityJob.	9.1124-Current
Temp Renew Asset Line Item	Temporary object to store renew asset line item processed by RenewOpportunityJob.	9.1124-Current
Temp Rollup Data	Represents the dynamic rollup data for use in criteria.	9.902-Current
Totaling Group	Represents a totaling group.	6.674 - Current
Transfer Price	Represents association of source price like factory price with source cost like FES source cost.	9.1098 - Current
Usage Price Tier	Represents a single entry in a usage price tier.	6.674 - Current

Apttus Contract Management

Apttus CLM Object	Purpose of the Object	Available in Versions
Account Hierarchy	This defines a single instance of an account hierarchy relationship. It contains the leaf Id, the parent Id, the root Id, and the level.	8.292 - Current
Admin	Store admin preferences and metadata.	7.207 - Current
Agreement	Represents an agreement in the system.	7.207 - Current
Agreement Action Condition	Represents conditions to enable agreement actions.	8.301 - Current
Agreement Clause	Represents a clause associated with an agreement.	7.207 - Current
Agreement Document	Links to agreement documents that are not stored in Apttus / Salesforce repository. These could be URLs to files stored on a file server or document management system behind the client's firewall.	7.207 - Current
Agreement Document Output Format	Holds default output format preferences by user profile and agreement type for use in the agreement document generation process.	7.207 - Current
Agreement Explorer	A reusable, unique combination of Field Sets and Filter Sets.	8.292 - Current
Agreement Hierarchy	This defines a single instance of an agreement hierarchy relationship. It contains the leaf Id, the parent Id, the root Id, and the level.	8.292 - Current

Apttus CLM Object	Purpose of the Object	Available in Versions
Agreement Line Item	Represents a product or service line item associated with an agreement.	7.207 - Current
Agreement Lock	Represents a lock on the agreement.	7.207 - Current
Agreement Protection	Specifies protection settings for agreements generated by the Apttus Contract Wizard and maintained by the Apttus Contract Author.	7.207 - Current
Agreement Rule	Specifies a rule used to evaluate various agreement actions such as submit request.	7.207 - Current
Agreement Rule Condition	Specifies a condition used to evaluate an agreement rule.	7.207 - Current
Agreement Term Exception	Term Exceptions associated with a given Agreement.	7.207 - Current
Async Merge Call	Holds asynchronous merge calls.	7.207 - Current
Content Event	Holds content events for subscribers to handle.	7.207 - Current
Cycle Time Field	Holds the fields monitored for cycle time data capture.	7.207 - Current
Cycle Time Field Data	Holds the data captured on fields for cycle time reporting.	7.207 - Current

Apttus CLM Object	Purpose of the Object	Available in Versions
Cycle Time Group	Holds the groups monitored for cycle time data capture.	7.207 - Current
Cycle Time Group Data	Holds the data captured on groups for cycle time reporting.	7.207 - Current
Doc Assembly Component	Represents a single component in a document assembly rule.	7.210 - Current
Doc Assembly Rule	Represents a single rule in a document assembly ruleset.	7.210 - Current
Doc Assembly Ruleset	Represents a rule set that contains a list of rules to determine the sections to include in a dynamic template.	7.210 - Current
Document Agreement Clause	Agreement clauses in an agreement document.	8.350 - Current
Document Version	Represents a document version.	8.270 - Current
Document Version Detail	Holds the version details of a document.	8.270 - Current
Formula Field (Comply)	Represents a formula field used in agreement rules.	7.207 - Current

Merge Event	Holds document merge events for subscribers to handle.	7.207 - Current
Merge Event Detail	Holds document merge event details.	8.270 - Current
Query Template	Query Template for building SOQL queries and filters.	7.207 - Current
Query Template Filter	Filter expression used in a query.	7.207 - Current
Query Template Qualifier	Filter used to select a query template.	7.207 - Current
Related Agreement	To identify the different types of relationships between agreements.	7.207 - Current
Retention Policy	Holds object retention policies.	7.207 - Current
Search Filter (Comply)	Represents a search filter.	7.207 - Current
Temp Object (Comply)	A container for temporary objects and attachments.	7.207 - Current

Apttus CLM Object	Purpose of the Object	Available in Versions
Template	Contract language templates available in the system. These templates specify guidance, language, local settings, and include Word attachment templates representing the bulk of actual contracts that can be used when creating the first draft. Templates can include standard and non-standard language.	7.207 - Current
Template Clause Reference	Represents a clause reference in a template or a library.	7.207 - Current
Template Clause Reference Version	Holds Template Clause References for this version.	8.326 - Current
Template Datasource Filter	Represents a data source filter associated with a template.	7.207 - Current
Template Dynamic Section	Represents a dynamic section in a template.	7.207 - Current
Template Dynamic Section Version	Holds the template dynamic section details for a given template version.	8.326 - Current
Template Version	Holds various versions of this template and is a Master-detail to Template.	8.326 - Current
Term Exception	Term Exceptions master table.	7.207 - Current

Wizard	A runtime instance of the wizard design.	8.299 - Current
Wizard Design	Holds all the wizard design records.	8.299 - Current
Wizard Input Control	The individual question, instruction statements, and associated field attributes.	8.299 - Current
Wizard Rule	Individual rules which are applied within a ruleset.	8.299 - Current
Wizard Ruleset	A collection of rules that are applied to input control and steps. The system determines a ruleset at the level of input control expression formula, determine focus objects, record type, and step navigation rules.	8.299 - Current
Wizard Runtime Input	Input Control instance within a runtime wizard.	8.299 - Current
Wizard Step	The step that is designed and used in Wizard designs.	8.299 - Current

Apttus Proposal Management

Apttus Proposal Management Object	Purpose of the Object	Available in Versions
Document Collate Info		8.111 - Current
Proposal Document Output Format	Holds default output format preferences by user profile and quote /proposal type for use in the quote/proposal document generation process.	7.72 - Current
Proposal Line Item	Represents a product or service line item associated with a quote /proposal	7.72 - Current
Proposal Request	Approval requests for a proposal	7.72 - Current
Quote /Proposal	Quote/Proposal object which holds dates and line items, many quotes/proposals may be related to a single opportunity, a quote /proposal may have many detail lines	7.72 - Current

Apttus Quote/Proposal-Configuration Integration

Apttus Quote/Proposal-Configuration Integration Object	Purpose of the Object	Available in Versions
Proposal Adjustment Line Item	Represents an adjustment to the price of a product /service.	6.99 - Current
Proposal Footnote	Represents the footnote available in a quote /proposal.	6.99 - Current
Proposal Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the quote/proposal.	9.139 - Current

Apttus Quote/Proposal-Configuration Integration Object	Purpose of the Object	Available in Versions
Proposal Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the quote/proposal.	9.134 - Current
Proposal Location	Represents the location available in a quote /proposal.	9.138 - Current
Proposal Product Attribute Value	Represents the attribute values for a product class in a proposal line item.	6.99 - Current
Proposal Rollup Data	Represents the dynamic rollup data for use in criteria.	9.134 - Current
Proposal Summary	Represents the summary of a product or service line item in a proposal.	6.99 - Current
Proposal Summary Group	Represents a summary of line items associated with a quote or proposal.	6.99 - Current
Proposal Tax Breakup	Represents the tax breakup of a product/service.	8.131 - Current
Proposal Usage Price Tier	Represents a single entry in a quote/proposal usage price tier.	6.99 - Current

Apttus Contract-Configuration Integration

Apttus Contract-Configuration Integration Object	Purpose of the Object	Available in Versions
Agreement Adjustment Line Item	Represents an adjustment to the price of a product /service.	6.63 - Current

Apttus Contract-Configuration Integration Object	Purpose of the Object	Available in Versions
Agreement Footnote	Represents the footnote available in an agreement.	6.63 - Current
Agreement Incentive Benefit Data	Represents a single benefit data row associated with the incentive benefit for the agreement.	9.74 - Current
Agreement Incentive Limit Data	Represents a single limit data row associated with the incentive limit for the agreement.	9.74 - Current
Agreement Product Attribute Value	Represents the attribute values for a product class in an agreement line item.	6.63 - Current
Agreement Rollup Data	Represents the dynamic rollup data for use in criteria.	9.74 - Current
Agreement Summary	Represents the summary of a product or service line item in an agreement.	6.63 - Current
Agreement Summary Group	Represents a summary of line items associated with an agreement.	6.63 - Current
Agreement Tax Breakup	Represents the tax breakup of a product/service.	9.74 - Current
Agreement Usage Price Tier	Represents a single entry in an agreement usage price tier.	6.63 - Current

Profile Settings and Security

As an administrator, you can create or modify permissions for users to access the Apttus CPQ objects and its fields. Below is a representative list of access configuration for various user types.

Profile	Apttus Administrator	Sales User	Comply User	Legal
Custom App Settings				
Apttus Contract Management	Yes	Yes	Yes	Yes
Apttus Order Management	Yes	Yes	No	No
Apttus Proposal Management	Yes	Yes	No	No
Apttus Product Setup	Yes	Yes	No	No
Apttus Pricing Setup	Yes	Yes	No	No
Apttus Batch Updates	Yes	Yes	No	No
Apttus Incentive Setup	Yes	Yes	No	No
Standard Tab Settings				

Profile	Apttus Administrator	Sales User	Comply User	Legal
Accounts	Use Existing Setting	Use Existing Setting	Use Existing Setting	Use Existing Setting
Contacts	Use Existing Setting	Use Existing Setting	Use Existing Setting	Use Existing Setting
Dashboards	Use Existing Setting	Use Existing Setting	Use Existing Setting	Use Existing Setting
Reports	Use Existing Setting	Use Existing Setting	Use Existing Setting	Use Existing Setting
Apttus Custom Tab Settings				
About Proposal Management	Default On	Default On	Tab Hidden	Tab Hidden
Admin	Default On	Tab Hidden	Tab Hidden	Tab Hidden
Agreement Document Output Formats	Default On	Default On	Tab Hidden	Tab Hidden
Agreement Protection	Default On	Tab Hidden	Tab Hidden	Tab Hidden

Profile	Apttus Administrator	Sales User	Comply User	Legal
Agreement Rules	Default On	Tab Hidden	Tab Hidden	Tab Hidden
Agreements	Default On	Tab Hidden	Tab Hidden	Tab Hidden
Attribute Maintenance	Default On	Default On	Tab Hidden	Tab Hidden
Bundle Maintenance	Default On	Default On	Tab Hidden	Tab Hidden
Categories	Default On	Default On	Tab Hidden	Tab Hidden
Category Maintenance	Default On	Default On	Tab Hidden	Tab Hidden
Charge Groups	Default On	Default On	Tab Hidden	Tab Hidden
Charge Types	Default On	Default On	Tab Hidden	Tab Hidden
Config Settings	Default On	Default On	Tab Hidden	Tab Hidden
Constraint Rules	Default On	Default On	Tab Hidden	Tab Hidden
Constraint Rule Maintenance	Default On	Default On	Tab Hidden	Tab Hidden

Profile	Apttus Administrator	Sales User	Comply User	Legal
CPQ Console	Default On	Default On	Tab Hidden	Tab Hidden
Criteria Maintenance	Default On	Default On	Tab Hidden	Tab Hidden
Cycle Time Groups	Default On	Tab Hidden	Tab Hidden	Tab Hidden
Field Expressions	Default On	Default On	Tab Hidden	Tab Hidden
Formula Fields	Default On	Default On	Tab Hidden	Tab Hidden
Formula Fields (Comply)	Default On	Tab Hidden	Tab Hidden	Tab Hidden
Frequency Conversion Rates	Default On	Default On	Tab Hidden	Tab Hidden
Guided Selling Rules	Default On	Default On	Tab Hidden	Tab Hidden
Incentive Groups	Default On	Default On	Tab Hidden	Tab Hidden
Incentive Rollups	Default On	Default On	Tab Hidden	Tab Hidden
Incentives	Default On	Default On	Tab Hidden	Tab Hidden

Profile	Apttus Administrator	Sales User	Comply User	Legal
Price Dimensions	Default On	Default On	Tab Hidden	Tab Hidden
Price Lists	Default On	Default On	Tab Hidden	Tab Hidden
Price Rulesets	Default On	Default On	Tab Hidden	Tab Hidden
Product Attribute Groups	Default On	Default On	Tab Hidden	Tab Hidden
Product Constraints	Default On	Dafault On	Tab Hidden	Tab Hidden
Product Default Rules	Default On	Default On	Tab Hidden	Tab Hidden
Product Filter Maintenance	Default On	Default On	Tab Hidden	Tab Hidden
Product Groups	Default On	Default On	Tab Hidden	Tab Hidden
Proposal Document Output Formats	Default On	Default On	Tab Hidden	Tab Hidden
Proposals	Default On	Default On	Tab Hidden	Tab Hidden
Query Templates	Default On	Tab Hidden	Tab Hidden	Default Off

Profile	Apttus Administrator	Sales User	Comply User	Legal
Retention Policies	Default On	Tab Hidden	Tab Hidden	Tab Hidden
Search Filters (CPQ)	Default On	Default On	Tab Hidden	Tab Hidden
Templates	Default On	Tab Hidden	Default Off	Default Off
Term Exceptions	Default On	Default On	Tab Hidden	Tab Hidden
Custom Object Permissions				
Account Billing Summary	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Account Locations	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Adjustment Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Admin	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreements	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Agreement Adjustment Line Items				
Agreement Clauses	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Documents	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Document Output Formats	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Footnotes	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Locks	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Price Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Price Tiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Product Attribute Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
			Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Agreement Protection	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete		
Agreement Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Rule Conditions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Summary	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Summary Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Term Exceptions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Agreement Usage Price Tiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Applied Expression Infos	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Applied Rule Action Infos	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Applied Rule Infos	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Asset Attribute Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Asset Attribute Value History	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Asset Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Asset Line Item History	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Asset Transaction History	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Asset Usage Price Tiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Asset Usage Price Tier History	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Async Merge Calls	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Attribute Value Matrix	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Attribute Value Matrix Entries	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Batch Jobs (CPQ)	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Batch Job (CPQ) Parameters	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Billing Plans	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Billing Plan Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Billing Preferences	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Bundle Components	Read, Create, Edit, Delete, View All, Modify All	Read	Read	Read
Categories	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Category Hierarchies	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Charge Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Charge Group Members	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Charge Types	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Collaboration Requests	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Constraint Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Constraint Rule Actions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Constraint Rule Action Expressions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Constraint Rule Conditions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Content Events	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Custom Messages	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Cycle Time Fields	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Cycle Time Field Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Cycle Time Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Cycle Time Group Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Doc Assembly Components	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Doc Assembly Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Doc Assembly Rulesets	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Document Versions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Document Version Detail	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
External Order Adjustment Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
External Order Summary	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
External Order Summary Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Features	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Feature Sets	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Field Expressions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Footnotes	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Formula Fields (Comply)	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Formula Fields (CPQ)	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Frequency Conversion Rates	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
G/L Account Map Entries	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
G/L Account Map Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Guided Questions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Guided Search Rule Entries	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Guided Search Rule Filters	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Guided Selling Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Help Docs	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Incentives	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Incentive Benefit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Incentive Coupons	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Incentive Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Incentive Limits	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Incentive Limit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Licenses	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Line Item Rollups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Merge Events	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Merge Event Details	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Milestones	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Orders	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Order Adjustment Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Incentive Benefit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Incentive Limit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Pocket Adjustment Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Product Attribute Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Rollup Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Tax Breakups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Order Usage Price Tiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Payment Terms	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Permission Set Relationships	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Price Breakups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price Dimensions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price Lists	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price List Categories	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price List Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price Matrices	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price Matrix Entries	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price Rule Entries	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Price Rulesets	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Attributes	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Product Attribute Group	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Attribute Group Members	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Attribute Matrix Views	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Attribute Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Attribute Rule Actions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Attribute Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Classifications	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Configurations	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Constraints	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Constraint Entries	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Constraint Views	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Product Default Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Default Rule Filters	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Default Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Features	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Feature Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Filter Views	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Footnotes	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Group Members	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Hierarchy Views	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Information	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Product Option Components	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Option Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Product Option Prices	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Adjustment Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Document Output Formats	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Footnotes	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Incentive Benefit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Incentive Limit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Location	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Product Attribute Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Proposal Requests	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Rollup Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Summary	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Summary Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Tax Breakups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposal Usage Price Tiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Query Templates	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Query Template Filters	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Query Template Qualifiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Proposals	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Related Account Locations	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Related Agreements	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Related Asset Line Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Related Incentives	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Related Price List Items	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Related Products	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Retention Policies	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Revenue Recognition Policies	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	-	-
Rollup Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Saved Searches	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Search Attribute Values	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Search Filters (Comply)	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Search Filters (CPQ)	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Summary Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Tax Breakups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Tax Certificates	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Tax Codes	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Temp Display Columns	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Temp Filters	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Temp Incentive Benefit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Temp incentive Limit Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Temp Objects	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Temp Objects (CPQ)	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Temp Renews	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Temp Rollup Data	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Templates	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Template Clause References	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Template Datasource Filters	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Template Dynamic Sections	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Term Exceptions	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Totaling Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Usage Price Tiers	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Designs	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Input Controls	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Profile	Apttus Administrator	Sales User	Comply User	Legal
Wizard Input Control Runtime	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Input Groups	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Rules	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Rulesets	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Ruleset Runtime	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Runtime	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Steps	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read
Wizard Step Runtime	Read, Create, Edit, Delete, View All, Modify All	Read, Create, Edit, Delete	Read	Read

Glossary of CPQ terms

A

Term	Description
Administrators	Individuals who are responsible for maintaining and mapping product catalogs and pricing.
Attribute-Based Pricing	The price is determined by factoring in attributes of a product.
Attribute Groups	Attributes associated to products
Attributes	Are used to associate certain features to a product. Examples of common attributes are color, size, weight, and more.

B

Term	Desscription
Base Extended Price	This is derived by Base Price X Quantity X Term.
Batch Job (CPQ)	Represents an individual batch job.
Bundle within a Bundle	Apttus allows you to create multi-layered products or “bundles within bundles.”
Bundled Products	Represent a combination of standalone products that offer added value to the customer while increasing overall sales.

Term	Desscription
Bundled Services	Services sold together, as a package, rather than separately.

C

Term	Description
Categories	Categories are high-level logical groupings of products. Their grouping affects the way the end user sees them for selection in a product catalog. A category is created for browsing products in the selection pages or they can be created for creating options groups for a bundled product. Products are associated to a category through a category hierarchy. Category hierarchies are maintained using the Hierarchy Manager. They are also used to total the product prices into logical groupings.
Category Hierarchy	Expresses the collection of subordinate categories into a superordinate category.
Clone	To replicate a field, record, template, etc.
Configuration Rules	Ensures an error-free build or customization, even for inexperienced sales reps.
Configure Products	Button on Quote/Proposal record that allows end users to start configuring a quote.
Constraint Rule Action	Captures information about rule actions, which are only applied when the rule conditions are satisfied.
Constraint Rules	Drive automatic inclusion, exclusion, recommendation or replacement of products selected on the catalog page. Each Constraint Rule is composed of three parts; the Rule Detail, the Rule Condition and the Rule Actions.
Contractual Pricing	Customer specific contractual prices are agreed and used for a defined period.

CPQ Console	The CPQ Console allows you to manage most product and pricing functions.
Customer Priority	Field on Account record that defines customer's priority or rating.

D

Term	**Description**
Discrete	The system only considers specific matrix values.

L

Term	**Description**
Line Item	Represents a product or service line item.

N

Term	**Description**
New UI	Apttus CPQ's revamped user interface which is designed with AngularJS framework (client side JavaScript framework that adds interactivity to HTML). The New UI is intuitive which leads to a better user experience.

O

Term	**Description**
Objects	A definition of a specific type of information you can store in Salesforce. Some objects are native to Salesforce (such as Contacts or Accounts) while others are specific to Apttus functionality (such as Proposals or Agreements).
Opportunity	Salesforce Standard object that allows you to create opportunity records.

Term	Description
Option Group	To combine product options and then associate that group to a bundled product.
Organization	A deployment of Salesforce that has a defined set of licensed users. Your organization includes all of your data and applications, and is separate from all other organizations.

P

Term	Description
Price Adjustments	Price rules that can be made in CPQ (example: Price Matrix and Price Ruleset).
Price Dimensions	Represents a price criteria dimension.
Price List Items	Each price list item includes the list price for the product including details of cost such as per unit, flat price, one time, recurring, and more. Each price list item also represents the different ways that a customer is charged for a product; for example: license fees, implementation fees, etc. Price list items are categorized into Price Lists.
Price Lists	Containers of items that are grouped in a price list. A price list controls which products are visible to the end user. A price list contains several price list items; each linked to a product. A product can be set up with one or more price list items.
Price Matrix (Matrices)	These are an advanced pricing concept used to define tiered pricing paradigms, or complex pricing structures with multiple criteria. Common examples are pricing tiers for a product based on user count or particular customer or transactional dimensions.
	Pricing mechanisms that establish pricing that varies by time (ramp) or quantity (tiers).

Term	Description
Price Ramps and Price Tiers	
Price Rule	Represents a single rule in a price ruleset.
Price Rule Sets	Price Rulesets are a mechanism to allow particular families, categories or groupings of products to have either line item pricing adjustments applied or summary pricing adjustments applied. Typical examples of these are volume discounting rules or promotional pricing rules.
Product Attribute Group	Represents a product attribute group that contains attributes shared by products.
Product Attribute Value	Represents the attribute values for a product class.
Product Console	The product console helps administer product properties and associate products with other CPQ artifacts.
Product Family	Is a Salesforce standard field on the product record with configurable values. This construct allows you to create combinations of products with similar characteristics /qualities for use in a Price Rule.
Product Group	Is an Apttus custom object that creates a logical grouping of one or more product records. This construct allows you to create combinations of products with similar characteristics/qualities for use in a Price Rule.
Product Hierarchy	Defines the structure of the product catalog.

Term	Description
Product Option Group	Represents a product option group which is a member of the list of option groups associated with a product.
Products	A product or service that can be sold on its own as a standalone item, an option of other products, or as part of a bundled product. Apttus utilizes the standard Salesforce Products2 table to store product records and Apttus-specific fields on those records that manage the behavior of products.
Proposal	The Apttus custom object used to record information about a specific quote for a specific Opportunity and Account. The tab for the object is presented as Proposals and the record is stored as a Quote/Proposal.

Q

Term	Description
Quote	The Apttus custom object used to record information about a specific quote for a specific Opportunity and Account. The tab for the object is presented as Proposals: and the record is stored as a Quote/Proposal.

R

Term	Description
Records	A collection of fields that store information about a specific item of a specific type (represented by an object), such as a contact, an account, or an opportunity.

S

Term	Description
Selling Term	Field on line item object (column in cart) that defines length of time the seller will allow the buyer to pay for a product.

Term	Description
Standalone Products	Refers to a device or software that is self-contained, one that does not require any other devices or software to function.
Standalone Services	Refers to a service that is self-contained, one that does not require any other service to function.

T

Term	Description
Tiered Pricing Model	The longer a subscription term on a quote, the lower the monthly rate will be.

V

Term	Description
Validating	To check or prove the validity or accuracy of a value or instance.

Apttus Contact Support

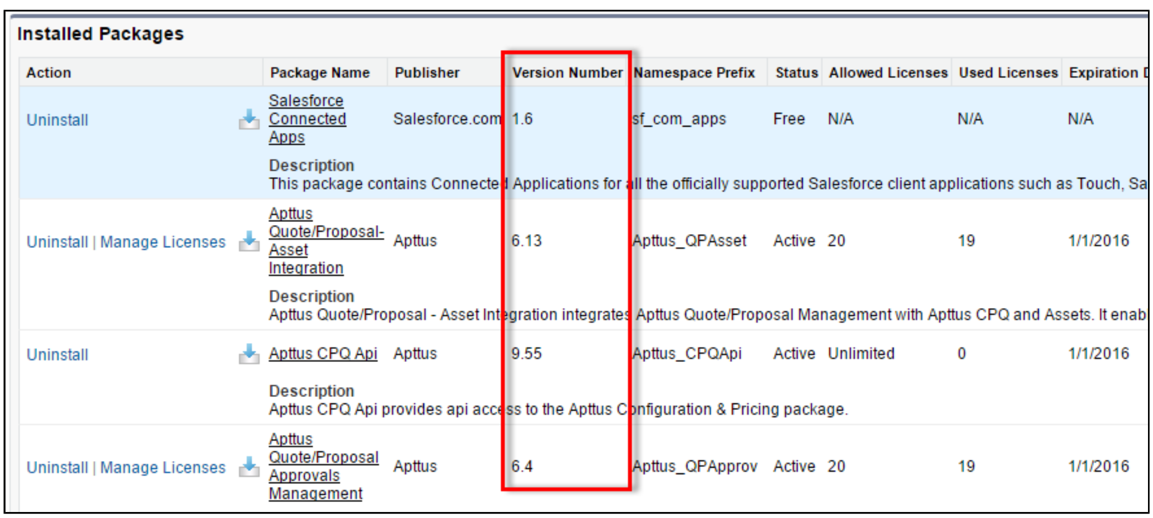
If you experience an issue with an Apttus product and need help, you can contact Apttus Support. Before you contact Apttus support, prepare a brief description of the problem you are experiencing. Additionally, to enable us to resolve your problem at the earliest, provide the following important information:

- What is the environment in which you are experiencing the problem: Sandbox or Production?
- How many users are affected?

Which product versions are installed?

To determine version numbers:

1. Go to Setup > App Setup > Installed Packages.
2. In the Installed Packages section, all the installed packages are displayed. You can find the version numbers in the Version Number column.



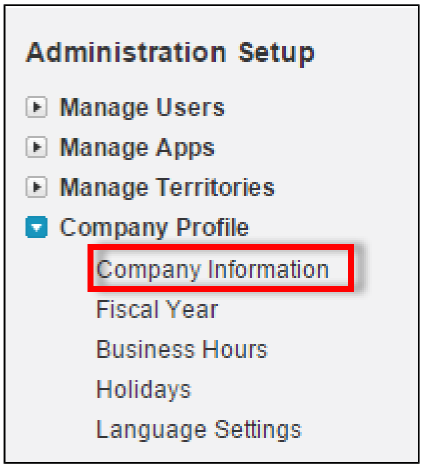
Installed Packages

Action	Package Name	Publisher	Version Number	Namespace Prefix	Status	Allowed Licenses	Used Licenses	Expiration D
Uninstall	Salesforce Connected Apps	Salesforce.com	1.6	sf_com_apps	Free	N/A	N/A	N/A
	Description This package contains Connecte Applications for all the officially supported Salesforce client applications such as Touch, Sa							
Uninstall | Manage Licenses	Apttus Quote/Proposal-Asset Integration	Apttus	6.13	Apttus_QPAsset	Active	20	19	1/1/2016
	Description Apttus Quote/Proposal - Asset Integration integrates Apttus Quote/Proposal Management with Apttus CPQ and Assets. It enab							
Uninstall	Apttus CPQ Api	Apttus	9.55	Apttus_CPQApi	Active	Unlimited	0	1/1/2016
	Description Apttus CPQ Api provides api access to the Apttus Configuration & Pricing package.							
Uninstall | Manage Licenses	Apttus Quote/Proposal Approvals Management	Apttus	6.4	Apttus_QPApprov	Active	20	19	1/1/2016

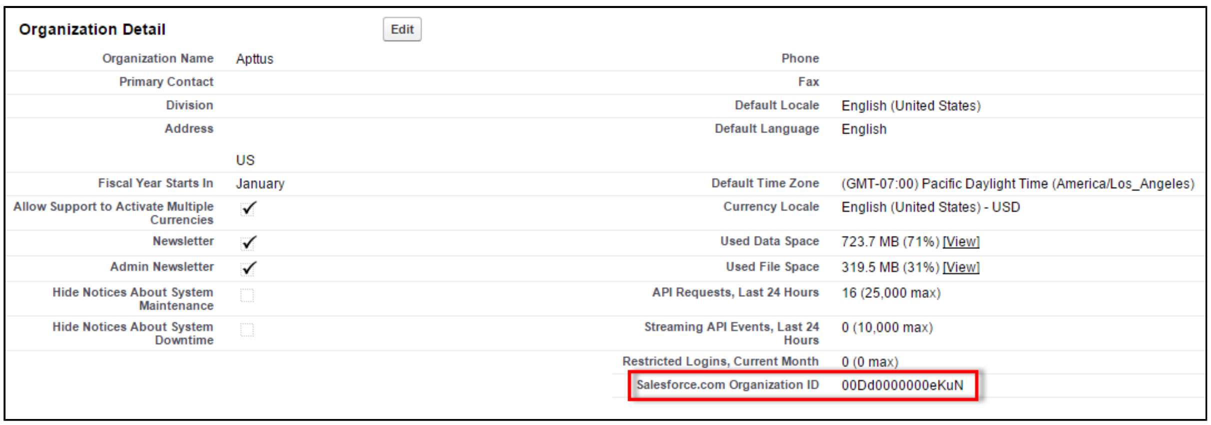
What is your Salesforce.com Organization ID?

To determine the Salesforce.com organization ID:

1. Go to Setup > Administration Setup > Company Profile > Company Information.



2. From the Organization Detail pane, provide the Salesforce.com Organization ID.



If you are having issues generating documents, what is your merge server end point?

To find the merge server end point:

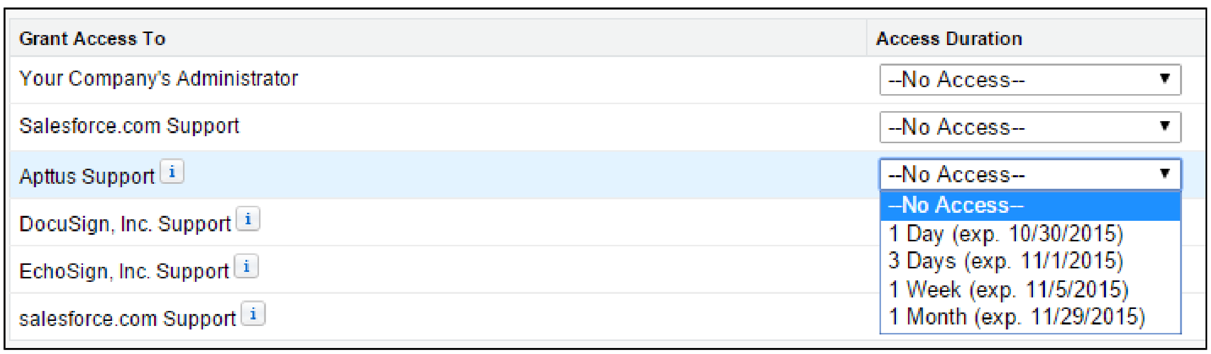
1. Go to Setup > App Setup > Develop > Custom Settings.
2. Click Manage for Comply System Properties.
3. Click System Properties.
4. The Merge Webservice Endpoint field displays the setting. The https://mergeserver.apttus.net:9876 portion of the setting is what will be helpful to customer support.

Grant Login Access of the affected user and an administrator.

To grant login access:

1. Go to Setup > Personal Setup > My Personal Information > Grant Login Access.

2. From the Apttus Support picklist, select an option for access duration.



3. Click Save.

Frequently Asked Questions

Why do I get the ‘An internal server error has occurred’ message when I click Configure Products, Go to Pricing, or Reprice?

This is a Salesforce error. Contact Salesforce Support for further assistance.

Why can’t I approve requests using email notifications?

1. Make sure that the email id saved in the email reply template on the visual force page is the same as the one saved in **Custom Settings** > **Email Services**.
2. Make sure that the Approver is a registered User with a valid Apttus license and then complete the following procedures.
    a. Click **Develop** > **Email Services** to create a new email service using ApprovalEmailHandler and generate an email address.
    b. Click **Develop** > **Custom Settings** > Approval System Properties and enter values for the fields described in the following table.

Field	Value
Enable Email Approval Response	Select the check box.
Email Approval Service Address	Enter the approver’s email address.

    c. On the Visual force email template (that is sent during approval submission) and enter the email address you generated in step 2.a, in the **Reply To** attribute of the template.
    d. Make sure that the template has the following fields.
        i. **Approval Request ID**
        ii. **Approval User ID**
    e. If you use a Custom VF Email Template, make sure the Apttus standard Merge fields (Approval ID and User ID) are NOT defined in Table format in code in Custom VF template or associated component. If these merge fields are in a Table format then Email to Approval will not work.

Note

Apttus does not support the system fields (Approval ID and User ID) in tabular format in the Custom VF email template code or the associated component code.

To resolve this, you must remove the Apttus standard Merge fields (**Approval ID** and **User ID**) from table and insert them separately in the following format.

Approval Request ID: {!relatedTo.ID}
Approval User ID: {!relatedTo.Apttus_Approval__Assigned_To_Id__c}

How can I resolve a hung page when I try to check-in or check-out any document on Apttus Author?

1. Make sure your profile has permission to Read, Create, Edit, and Delete the Template Object.
2. Make sure your profile has access to the following Visual force pages.
   a. Apttus.BrowseTemplates
   b. Apttus.BrowseAgreements
3. Log in with your credentials and paste the following link in the browser's URL. https://<instance>.salesforce.com/apex/Apttus__BrowseTemplates?callerMode=Checkout

Why do I get the 'An internal server error has occurred' message when I click Configure Products, Go to Pricing, or Reprice?Why does my sandbox instance redirect to my production instance whenever I refresh the page?

This is a Salesforce error. Contact Salesforce Support for further assistance.

Why do I get the 'An internal server error has occurred' message when I click Configure Products, Go to Pricing, or Reprice?

When you refresh the sandbox, the app copies all the data and components from the production instance to the sandbox instance.

The following table lists the applications and corresponding custom settings you must change or verify to resolve this issue.

Note

The instructions in the following table direct you to change settings in your sandbox instance. If you face the same problem on your production instance (switching to the sandbox instance when you refresh), use the same procedure but be sure to enter the URL for the production instance.

App	Settings
Apttus CPQ	1. Click **Setup** > **App Setup** > **Develop** > **Custom Settings** > **Config System Properties** > **Edit System Properties.** 2. Change the Production instance URL to the sandbox instance URL.
Apttus CLM	1. Click **Setup** > **App Setup** > **Develop** > **Custom Settings** > **Comply System Properties** > **Edit System Properties.** 2. Change the Production instance URL to the sandbox instance URL.
Apttus Approval Management	1. Click **Setup** > **App Setup** > **Develop** > **Custom Settings** > **Approval System Properties** > **Edit System Properties.** 2. Change the Production instance URL to the sandbox instance URL.

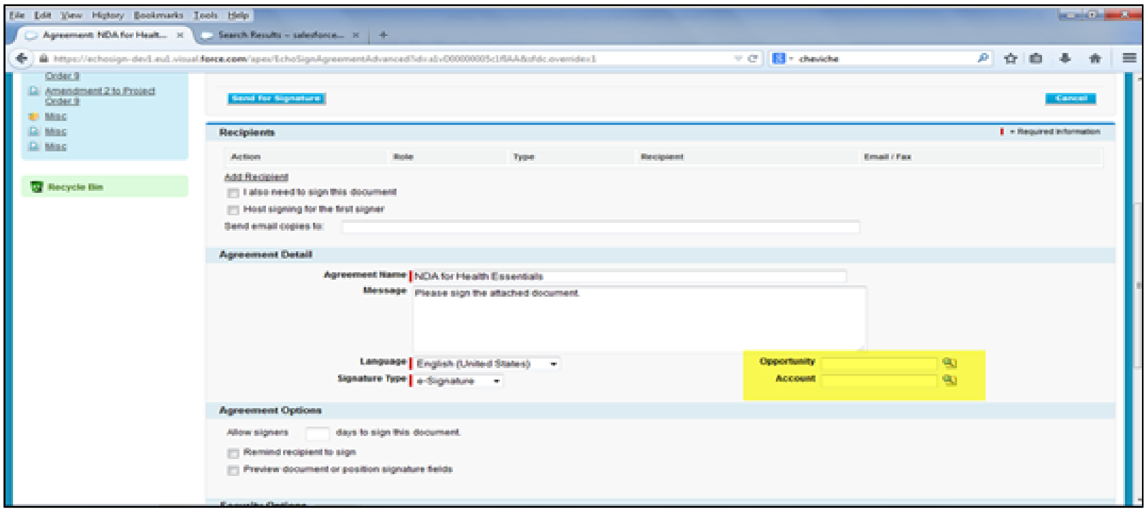
What Object references should I set to the instance of an Object so I can log in to X-author for contracts?

If you have this question, you receive the following message when you try to log in to X-Author for Contracts.

This error is not because there are Object permission issues in CPQ, in spite of the wording, this is a Time Zone error. To resolve this issue, change the time zone in the user profile settings to match the time zone in the Org settings.

How do I get the EchoSign Send page to auto-populate the Account and Opportunity fields?

The Account and Opportunity fields on the EchoSign Send page are lookup fields and the Apttus Integration Adaptor cannot auto-populate these fields.



To auto-populate these fields, employ one of the following work-around.

1. Create a custom trigger to auto-populate the **Account** and **Opportunity** fields from the Agreement. Because these are lookup fields, you cannot use workflow rules.
2. Click **Custom Settings** > **EchosignSettings** > **Edit**.
3. On the EchoSign settings page, uncheck the following check boxes.
   - **Disable Opportunity Lookup**
   - **Disable Account Lookup**
4. Click **Save.**

How do I enter values for custom fields on the Price List Item page?

The Price List Item detail page is a managed, custom visual force page where you cannot enter values for custom fields.

To be able to enter values on the Price List Item detail page, you must create a button of type URL that sales users can access to define values for custom fields. Use the following URL when you create this button.


```
{!URLFOR($Action.Apttus_Config2__PriceListItem__c.Edit,
Apttus_Config2__PriceListItem__c.Id,[retURL=URLFOR($Action.
Apttus_Config2__PriceListItem__c.View,
Apttus_Config2__PriceListItem__c.Id)],true)}
```

How is the Base Price calculated when I renew or change an asset with custom pricing?

You must use custom pricing for an asset if you want to override the defined current price of the asset.

For example, you are a Sales Representative for a software company and you receive a request to renew a software subscription. The price of the original order is $100.00. The current price of the product is 150.00. You can process the renewal for the original price if you enable Custom Pricing.

If you have enabled Custom Pricing for an asset, Apttus CPQ uses the Net Price of the asset to calculate the asset price. The formula is ***Net Price/Quantity*Term***.

When I create any template, why do I see the following error?

```
Error: Script Thrown Exception
```

Your profile does not have Edit permissions and so you do not have access to the following custom fields for the Agreement Object.

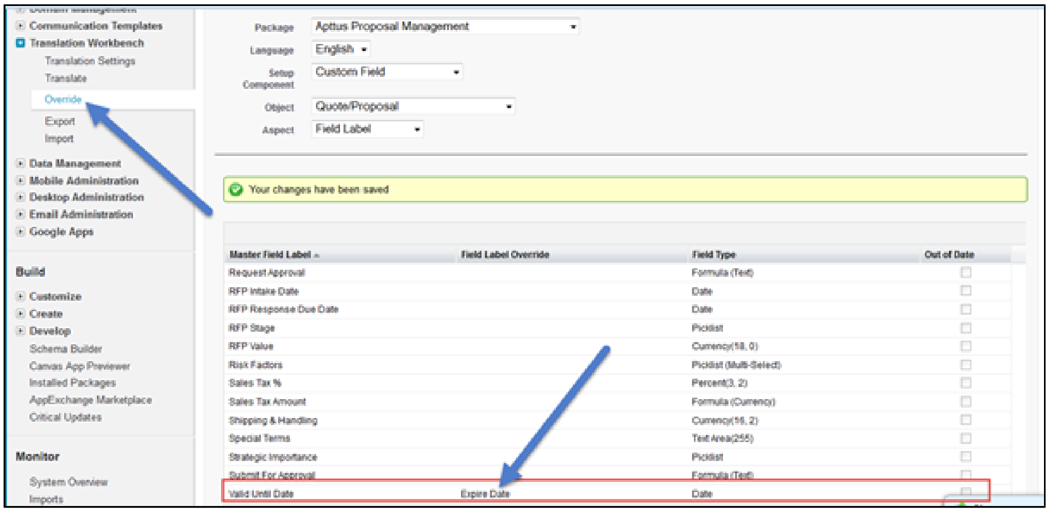
- Status Category
- Status
- Workflow Trigger Generated Agreement

Ask your system administrator to enable Edit permissions for the Agreement Object.

How do I change the label for an Apttus Managed field?

To change the label of an Apttus Managed field,

1. Click **Setup** > **Translation Workbench** > **Override**.



2. Double click in the **Field Label Override** column to enter a new value.
3. Click **Save**.

How do I clone a User Record?

Cloning a User record is not a Salesforce function. You must create a custom link along with formula to clone a user record.

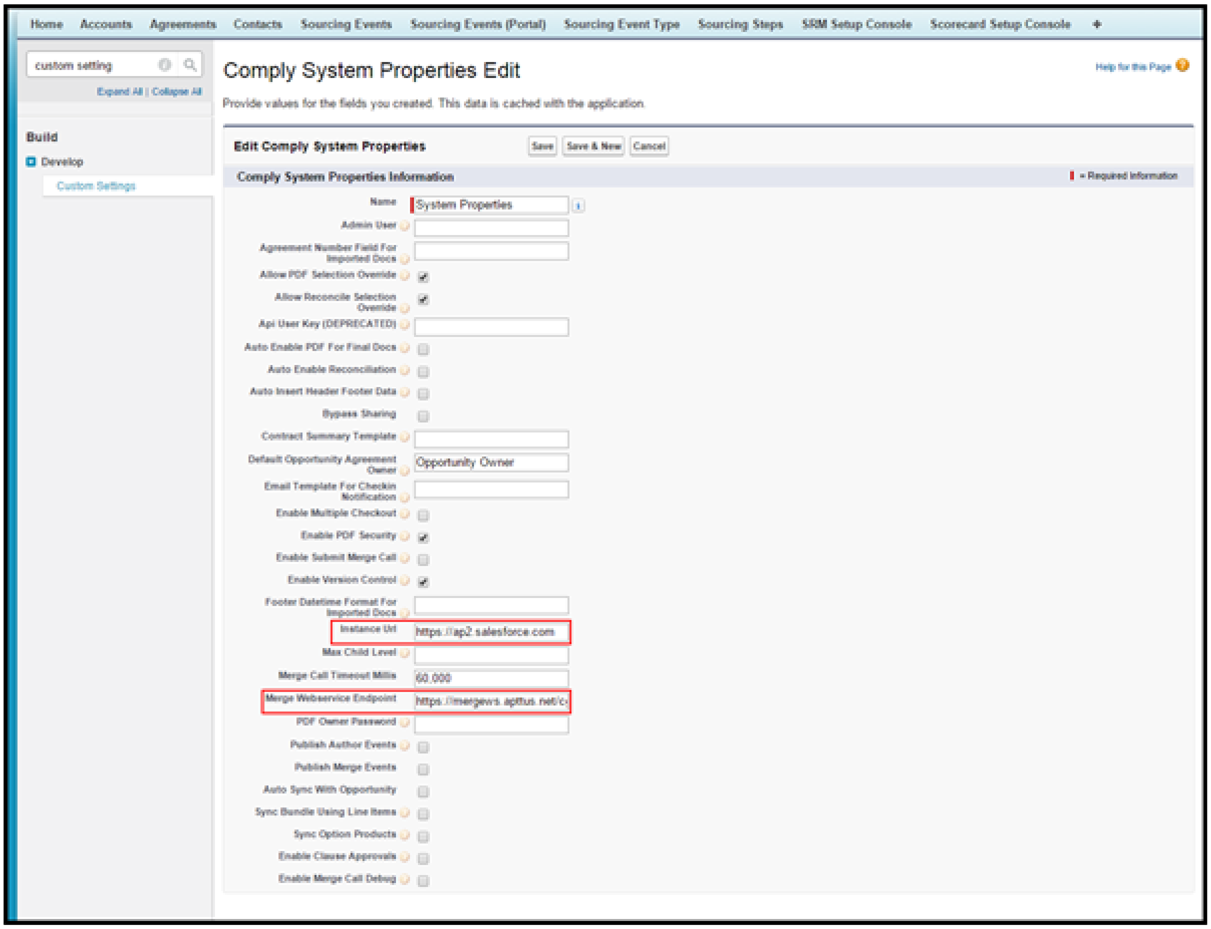
Create a custom link with the following formula.

```
/{!User.Id}/e?clone=1&retURL=%2F{!User.Id}
&name_firstName=&name_lastName=&Alias=&Em
```

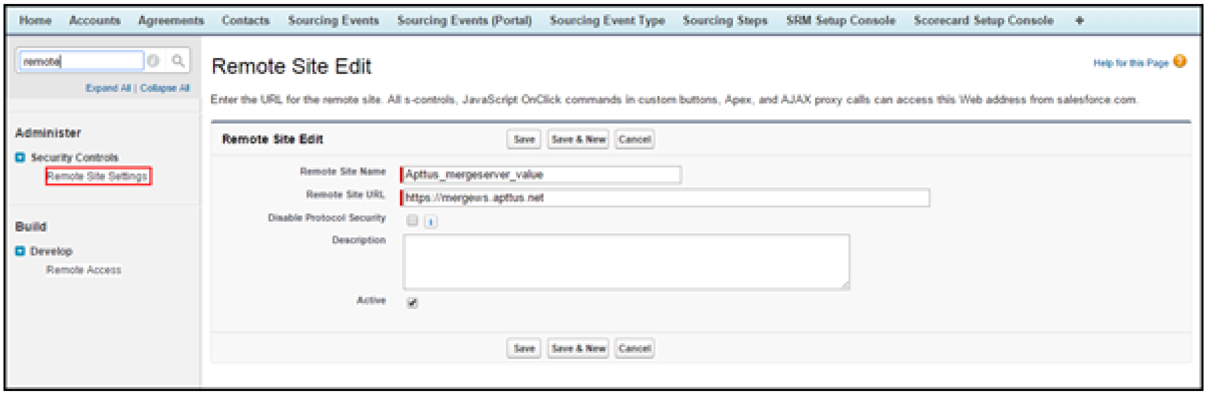
How do I add the Apttus Generate document functionality on any custom object?

You can add the Apttus Generate document functionality on any custom object by creating a formula on the custom object **Edit** .

1. Click **Setup > Build > Develop > Custom Settings > Comply System Properties > System Properties > Edit**.



2. Define values for the following fields.
    - **Merge Webservice Endpoint**
    - **Instance URL**
3. Click **Save**.
4. Click **Setup > Administer > Security Controls > Remote Site Settings > New Remote Site**.



5. Define values for the following fields.
   - **Remote Site Name**
   - **Remote Site URL**(your merge server URL)
6. Create a **Record Type** (at least on record type. For example, **Rec Type**) on the custom object, where you want to add the Generate document functionality.
7. Add that record type to the Agreement Types field (*Apttus__APTS_Template__c. Apttus__Agreement_Types__c*) as a picklist value.
8. Add that record type to the Type field (*Apttus__APTS_Template__c. Apttus__Type__c*) as a picklist value.
9. Create a Formula field for the **Generate** button on your custom object with the below expression.

```
("/apex/Apttus__DocumentGenerate?id="&Id&"&action=Generate_Docum
ent&templateType=RecTyp",IMAGE("/resource
/Apttus__Button_Generate","Generate"),"_self")
```

10. Click **Save**.

When I submit the agreement by clicking Submit Request button, why do I get the following error?

```
Error: Missing Target ObjectId with Template
```

You don’t have default contact access to the System Administrator Record.There is an Admin Entry created and defined as System Administrator as a Default Email Contact Name.

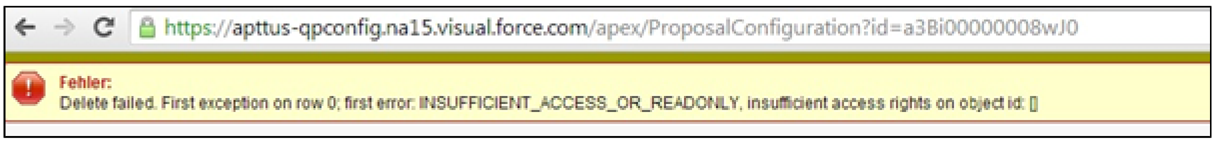
APTS_DefaultEmailContactName: Symantec Admin

Whenever you submit the Agreement record, the system looks for System Administrator contact. But, the OWD settings is defined as Private for Contact object, so this contact record doesn't have access to the concerned user. So the concerned user must have the System Administrator contact access to submit the records.

> **Note**
>
> The System Administrator has all record access automatically. So, Admin should not get any issue, but the users don't have access to all records.

Your System Administrator must provide you with access to **System Administrator** record sharing.

When I click the Configure Products button, why do I get the following error?



You will need **Modify All** permission on Product Configurations object.

The Quote Owner or System Admin will be able to configure any products for a quote. If any other user other than quote owner will try to do the same, the system will now the user to configure the Quote and will get the above error.

Ask your System Administrator to provide you with **Modify All** permissions on Product Configurations object.

When I run the Bundle Maintenance, why do I get the following error?

```
Error: First error: Insert failed. First exception on row 0; first
error: NUMBER_OUTSIDE_VALID_RANGE, Depth: value outside of valid
range on numeric field: 10: [Apttus_Config2__Depth__c]
```

You get this error when you've already ran bundle maintenance earlier, and in the middle you abort this process. In our system, we maintain the maintenance jobs status in an object called Batch Job (Apttus_Config2__BatchJob__c). Once the status is stuck in Queued stage, the actual batch job is aborted, and you will face this problem.

Resolution:

When you receive the above error, after running Bundle Maintenance, perform the following steps to resolve records with product components where the option group is null in the **Product Option Component** object.

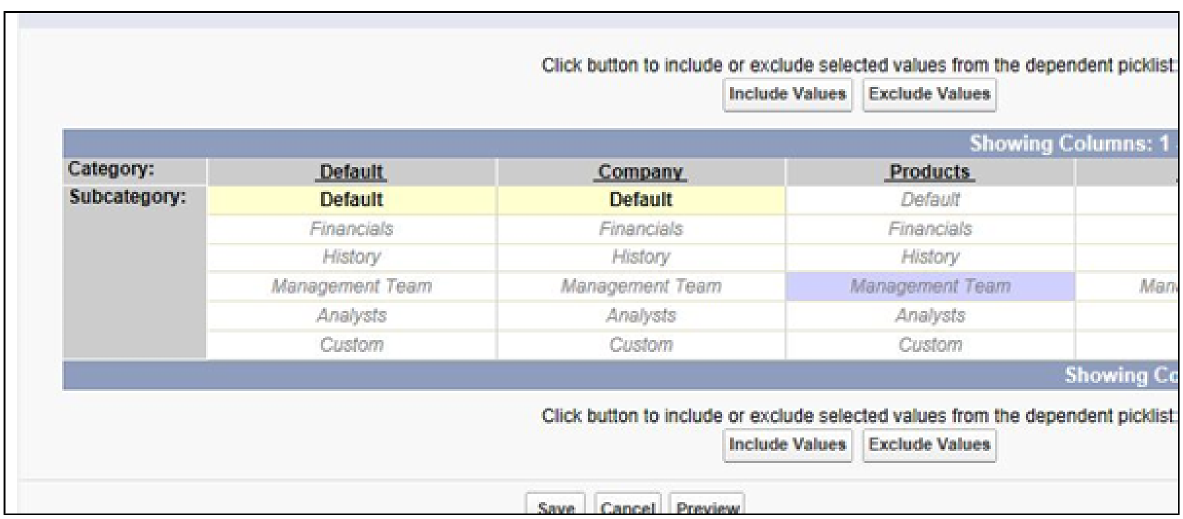
Run the following Query in the Developer Console. (a Salesforce System Administrator should be utilized or informed when performing this operation).

```
SELECT name, id, Apttus_Config2__ComponentProductId__c,
Apttus_Config2__RelationshipType__c,
Apttus_Config2__ParentProductId__c from
Apttus_Config2__ProductOptionComponent__c where
Apttus_Config2__ProductOptionGroupId__c = NULL
```

You will receive a list of returned IDs from the above Query. You must identify and delete the Product Component records without any option group assigned. Now, the Bundle Maintenance will execute properly.

How do I include the values in Subcategory field which are lost while saving a template in XAuthor V6.0?

1. Go to **Setup** > **Create > Objects.**
2. Search for **Template** object and click to open it.
3. Go to **Custom Fields and Relationships**.
4. Click **Category.**
5. Go to **Field Dependencies** and edit **Subcategory** to see the following screen.



Apttus Copyright Disclaimer

The documentation and/or software may provide links to Web sites and access to content, products, and services from third parties. Apttus is not responsible for the availability of, or any content provided by third parties. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Apttus is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Apttus is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

For additional resources and support, please visit https://apttusportal.force.com/community.

